NASA Technical Memorandum 83304

Production Version of the Extended NASA-Langley Vortex Lattice FORTRAN Computer Program - Volume II - Source Code

Henry E. Herbert
John E. Lamar

April 1982

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

PRODUCTION VERSION OF THE

EXTENDED NASA-LANGLEY VORTEX LATTICE

FORTRAN COMPUTER PROGRAM

VOL. II SOURCE CODE

Henry E. Herbert, Computer Sciences Corporation

and

John E. Lamar, NASA - Langley Research Center

Hampton, Va. 23666

CONTENTS

1 ABSTRACT / SUMMARY

This document presents the source code for the latest production version, MARK IV, of the NASA - Langley Vortex Lattice Computer Program. All viable subcritical aerodynamic features of previous versions have been retained. This version extends the previously documented program capabilites to four planforms, 400 panels, and enables the user to obtain vortex-flow aerodynamics on cambered planforms, flow field properties off the configuration in attached flow, and planform longitudinal load distributions.

2 INTRODUCTION

The NASA - Langley Vortex Lattice FORTRAN Program (VLM) is designed to estimate the subsonic aerodynamic characteristics of up to four complex planforms. The concepts embodied in this program are mostly detailed in references 1,2 and 3; this document is intended to serve as an update to these references for users and computer specialists who have an interest in implementing this program on their computers. Basically, the VLM Program is a segmented program designed to run on the Control Data Corporation (CDC) computers with the NOS operating system. This program requires a run-time field length of approximately 130K (octal) words of memory, and uses the Langley Research Center

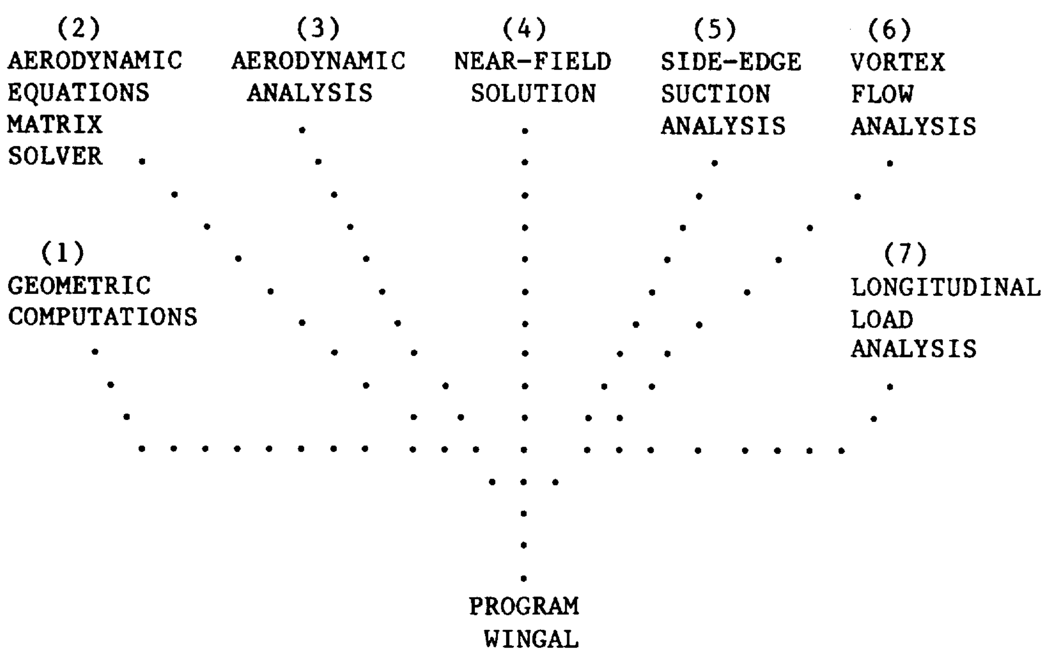
Graphics Output System in FORTRAN (LRCGOSF), along with numerous CDC routines that provide random access of mass storage files. The User's Guide associated with this program is listed as reference 4.

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

3 PROGRAM STRUCTURE

3.1 OVERVIEW

The VLM program consists of one FORTRAN program, 39 FORTRAN subroutines, seven labeled common blocks and nine files. The overall structure of the code can be represented as a tree diagram with seven major branches, as follows:



The main program, WINGAL, forms the "root node" to the tree. Each of the seven major branches, or nodes, has a specific function in the overall computational process, and some of these branches consist of more than one routine (this will be explained in the next section). The point of this tree diagram is that it constitutes the way the program is loaded into memory by the operating system, and hence, the way it executes for any given input set. The actual file of loader directives file based on this structure will be discussed in Section 4.

3.2 FUNCTION BY NODE

The root node consists of the main program (WINGAL) and four subroutines. The main program is used to declare all files and common blocks, as well as direct the overall processing done by this code. The four subroutines contained in this node are LOADING, FTLUP, INFSUB and READIN. LOADING is not called by any routine in VLM; its sole purpose is to force the loading of the CDC mass storage routines at this level so they are accessible to all higher level nodes that require them. FTLUP and INFSUB, however, are used by several of the higher nodes, and hence, must be in memory at all times. FTLUP is a linear interpolation routine that uses the CDC Fortran routine LOCF to determine the absolute location (address) of a variable. READIN is used to read in and print out the input data, with line numbers, for the user's reference. The common blocks declared in WINGAL serve as

the principal method of information transfer between the higher nodes; they are considered "global" and must also remain in memory at all times. The higher nodes in the tree are moved in and out of central memory by the loader when they are called by the main program, and this overlapping results in considerable reduction of central memory requirements. The function and associated routines of the higher nodes are as follows:

3.2.1 GEOMETRIC COMPUTATIONS

This node consists of two subroutines, GEOMETRY and PLNPLT. Subroutine GEOMETRY is called by WINGAL to determine, from the input data, the geometry of the configuration. GEOMETRY then makes a call to PLNPLT, which, in turn, produces the "printer plot" of the configuration.

3.2.2 AERODYNAMIC EQUATIONS MATRIX SOLVER

This node consists of six routines and performs the Given's Method of Matrix Solution to determine the circulation terms of the horseshoe vortices; i.e., these routines solve the matrix of the basic linear aerodynamic equations. The routines in this node are:

1. MATXSOL- Called by WINGAL, MATXSOL is the main routine in this node and generates the elements in the aerodynamic influence

coefficient matrix. MATXSOL then calls routine GIVENS to effect a solution.

2. GIVENS - Called by MATXSOL to partition the work storage arrays into rows and columns.

3. BLOCKR - Called by GIVENS to compute the size of, and the number of rows in each partition of the triangularized matrix.

4. TRIANG - Called by GIVENS to triangularize the augmented matrix using planar rotations.

5. SOLVER - Called by TRIANG to perform the back substitution on the matrix and overstore the results back onto mass storage.

6. BUFFIN - Called by TRIANG to transfer data into the work array from mass storage.

3.2.3 AERODYNAMIC ANALYSIS

This node consists of five routines and is used to compute the linear aerodynamic characteristics of the configuration. These routines are:

1. AERODYN - Called by WINGAL, AERODYN is the main routine in this node. AERODYN computes the linear aerodynamic lifting pressures and overall forces and moments.

2. FLOWFL - Determines the flow field characteristics off the wing. FLOWFL reads the field line definition when the flow field data is required. FLOWFL also uses routine FTLUP.

3. HEAPSRT - Called by FLOWFL to sort the X-Y values of the panel centers. FLOWFL compares these values to the X-Y locations of the flow line points in determining the relative position of the flow line with respect to any given planform. HEAPSRT uses a Heap Sort alogorithm, which is described in reference 5.

4. SIFT - Called by HEAPSRT to swap values in the sorting process.

5. CDICLS - Computes the far-field induced drag for simple configurations (wing-body) with no dihedral.

3.2.4 NEAR-FIELD INDUCED DRAG SOLUTIONS

This node consists of the single routine, CDRAGNF, (called by WINGAL) that computes the near-field induced drag values. CDRAGNF uses routine FTLUP in determining the near-field chord force properties.

3.2.5 SIDE-EDGE SUCTION ANALYSIS

This node consists of two routines, TIPSUCT and WRTANS. The principal routine, TIPSUCT, called by WINGAL, computes the side-edge force and values for $K_{V,se}$ for each planform. WRTANS is then called by TIPSUCT to compute the values of K_p and $K_{V,le}$ for each planform. If the configuration is cambered/twisted so that the option for vortex flow computation on warped wings is exercised, WRTANS is not called and K_p and $K_{V,le}$ are not calculated. In this situation, TIPSUCT makes numerous calls to the LRCGOSF routines for producing the graphics output. This section of TIPSUCT will have to be revised considerably by users if the VLM program is installed on any other computer. The LRCGOSF routines used by VLM are listed in the Appendix A.

3.2.6 VORTEX FLOW ANALYSIS

This node consists of the single routine, VORTEX, called by WINGAL, to perform the vortex flow analysis for cambered configurations. VORTEX also uses routine FTLUP.

3.2.7 LONGITUDINAL LOAD DISTRIBUTION

This node consists of 18 routines, the principal one being CNLONG, that called by WINGAL. These routines compute the longitudinal load distribution of the configuration. Most of them are associated with the Delta Cp or Net Cp surface interpolation from constant Y to constant X values. They are (by name only) as follows:

INTERP	SUTS
IQHSCV	CURV1N
IQHSD	CURV12
IQHSE	CURVI
IQHSF	CEEZ
IQHSG	TERMS
IQHSH	SNHCSH
UERTST	INTRVL
UGETIO	

The routines beginning with the letters "IQ" provide smooth curve fitting over randomly distributed data points. IQHSCV calculates the interpolating function that is a fifth degree polynomial, and is continuous and has continuous first order partial derivatives. This

technique is detailed in reference 6. Routines IQHSD through UGETIO are support routines for this process. Routines SUTS through INTRVL are used to evaluate the the integral of the spline approximation to a function of a single variable. These routines were adapted from current routines on the NASA - Langley Research Center FORTRAN Math Library which is documented in reference 7. This documentation is available from the NASA - LaRC Analysis and Computation Division (ACD) User Support Office.

3.3 FILES

There are nine files associated with the VLM program and these are as follows:

INPUT	TAPE6	TAPE20
OUTPUT	TAPE10	TAPE30
TAPE5	TAPE11	TAPE81

TAPE81 is equated to INPUT and TAPE6 to OUTPUT, and all formatted INPUT/OUTPUT is performed on these files. Routine READIN reads the input from TAPE81 and prints all of it except the title card over onto TAPE5. For the remainder of the program, TAPE5, serves as the primary input file. The remaining files are scratch files used by the program for auxillary storage. TAPE10 and TAPE11 are used in the matrix solver routines and are treated as random access backing store. The CDC routines READMS, WRITEMS, OPENMS, and CLOSEMS provide this capability.

TAPE10 is reused as a scratch file in the vortex analysis routine. There it is employed along with TAPE20 and TAPE30, and subsequently, TAPE30 is used to pass information from routine VORTEX back to routine TIPSUCT for the final coefficient reporting.

4 LOAD STRUCTURE

The tree diagram shown earlier can now be represented as follows:

```
GEOMTRY   MATXSOL   AERODYN   CDRAGNF   TIPSUCT   VORTEX   CNLONG
   .         .         .         .         .        .        .
   .         .         .         .         .        .        .
   ............................................................
                                 .
                                 .
                                 .
                              PROGRAM
                               WINGAL
```

where the nodes are main routines instead of functional descriptors. On CDC equipment, this method of loading is called Segmentation and is done by the CDC SEGRES Loader Program. The SEGRES program reads a directives file, shown in figure 1, and in effect structures the relocatable binary file into a set of movable pieces (here, each branch of the tree). When the main routine in these branches is called by WINGAL, the Loader brings into memory the entire branch, leaving the others out on mass storage. When a branch finishes processing and returns control back to the main program (WINGAL), that branch is moved back out to mass storage, and another one brought in and loaded into

the same space occupied by the previous one. Hence, the same space in memory is reused, which reduces the core requirements of the program. Complete details regarding the syntax of these directives and the actions of the SEGRES Loader Program are given in reference 8.

APPENDIX A

LRCGOSF ROUTINES USED BY VLM

PSEUDO
CALPLT
INFOPLT
NOTATE
PNTPLT

Documentation on these routines is given in reference 9, which is available from the NASA - LaRC ACD User Support Office.

APPENDIX B

LISTING OF VLM SOURCE CODE

```
      PROGRAM WINGAL(INPUT,OUTPUT,TAPE81=INPUT,TAPE6=OUTPUT,           WINGAL 2
     $         TAPE5,TAPE10,TAPE11,TAPE20,TAPE30)                      WINGAL 3
C                                                                      WINGAL 4
C                                                                      WINGAL 5
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,               ALL    2
     $    STA(4), TBLSCW(100), YYCP(4),                                ALL    3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),        ALL    4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                       ALL    5
C                                                                      ALL    6
      COMMON /TOTHREE/ CIR(400,2)                                      TOTHRE 2
C                                                                      TOTHRE 3
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD              THREFOR2
C                                                                      THREFOR3
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,   ONETHRE2
     $   ARTRUE, RTCDHT(4), CONFIG(2), NSSWSV(4),                      ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                              ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),     ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                         ONETHRE6
     $  YINNER(4), YOUTER(4)                                           ONETHRE7
C                                                                      ONETHRE8
      INTEGER CONFIG                                                   ONETHRE9
C                                                                      ONETHR10
      COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4), MAINONE2
     $  XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),      MAINONE3
     $  XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),   MAINONE4
     $  ZZ(25,4), ITIPCOD, ICAMTST                                     MAINONE5
C                                                                      MAINONE6
C                                                                      WINGAL11
C                                                                      WINGAL12
C              PRODUCTION  CODE                                        WINGAL13
C              MARK  IV  VERSION                                       WINGAL14
C              SEPTEMBER, 1981                                         WINGAL15
C                                                                      WINGAL16
C           PLEASE DIRECT ALL QUESTIONS, COMMENTS, ETC. TO:            WINGAL17
C              DR. JOHN E. LAMAR                                       WINGAL18
C              NASA - LANGLEY RESEARCH CENTER                          WINGAL19
C              HAMPTON, VA.  23666    M.S. 287                         WINGAL20
C                                                                      WINGAL21
C           SOFTWARE SUPPORT ON THIS VERSION PROVIDED BY:              WINGAL22
C              HENRY E. HERBERT                                        WINGAL23
C              COMPUTER SCIENCES CORPORATION                           WINGAL24
C              HAMPTON, VA. 23666                                      WINGAL25
C                                                                      WINGAL26
C                                                                      WINGAL27
C                                                                      NOTES  2
C             ---  NOTES TO THE USERS  ---                             NOTES  3
C                                                                      NOTES  4
C                                                                      NOTES  5
C        1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING              NOTES  6
```

```
C          EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF              NOTES   7
C          ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE                 NOTES   8
C          CALCULATED FROM CLDES = 100. AND CLDES = 1.0                NOTES   9
C                                                                      NOTES  10
C                                                                      NOTES  11
C        2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOM-   NOTES  12
C           MENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE  NOTES  13
C          MORE MEANINGFUL SIDE EDGE RESULTS                           NOTES  14
C                                                                      NOTES  15
C                                                                      NOTES  16
C                                                                      NOTES  17
C        3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO        NOTES  18
C           O. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE        NOTES  19
C           EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO      NOTES  20
C           1.0 IF THIS LOADING IS TO BE NON-ZERO                      NOTES  21
C                                                                      NOTES  22
C                                                                      NOTES  23
C                                                                      WINGAL29
      COMMON/CCRRDD/ TSPAN(4), TSPANA, KBIT, CTILDA, XTILDA, DISTALE   CCRRDD 2
C                                                                      CCRRDD 3
      DIMENSION INDEX(2)                                               WINGAL31
C                                                                      WINGAL32
C        VORTEX LATTICE AERODYNAMIC COMPUTATION                        WINGAL33
C            NASA-LRC PROGRAM NO. A2794                                WINGAL34
C                                                                      WINGAL35
C                                                                      WINGAL36
      CALL OPENMS(11,INDEX,1,0)                                        WINGAL37
C                                                                      WINGAL38
      CALL READIN                                                      WINGAL39
C                                                                      WINGAL40
      ICODEOF=TOTAL=0                                                  WINGAL41
 10   CALL GEOMTRY                                                     WINGAL42
      IF (M .EQ. -1) GO TO 70                                          WINGAL43
      IF(ICAMTST .EQ. 3) GO TO 70                                      WINGAL44
      IF (ICODEOF.GT.0) GO TO 70                                       WINGAL45
      IF (M .GT. 400) GO TO 40                                         WINGAL46
      NSW = 0                                                          WINGAL47
      DO 15 IT = 1,IPLAN                                               WINGAL48
      NSW = NSW + NSSWSV(IT)                                           WINGAL49
   15 CONTINUE                                                         WINGAL50
      IF (NSW .GT. 100) GO TO 30                                       WINGAL51
      ITSV=0                                                           WINGAL52
      DO 20 IT=1,IPLAN                                                 WINGAL53
      IF (AN(IT).LE.25.) GO TO 20                                      WINGAL54
      WRITE (6,100) IT,AN(IT)                                          WINGAL55
      ITSV=1                                                           WINGAL56
 20   CONTINUE                                                         WINGAL57
      IF (ITSV.GT.0) GO TO 60                                          WINGAL58
      GO TO 50                                                         WINGAL59
```

```
30    WRITE (6,90) NSW                                                  WINGAL60
      GO TO 60                                                          WINGAL61
40    WRITE (6,80) M                                                    WINGAL62
      GO TO 60                                                          WINGAL63
50    CALL MATXSOL                                                      WINGAL64
      CALL AERODYN                                                      WINGAL65
      IF (ITIPCOD .EQ. 3) CALL CNLONG                                   WINGAL66
      IF(CLDES.EQ.100.) GO TO 55                                        WINGAL67
      IF(PTEST.EQ.1..OR.QTEST.EQ.1..OR.ITIPCOD.EQ.2) GO TO 60           WINGAL68
      CALL CDRAGNF                                                      WINGAL69
      GO TO 57                                                          WINGAL70
   55 CALL VORTEX                                                       WINGAL71
C                                                                       WINGAL72
C     TEST FOR ERROR CONDITONS DETECTED IN VORTEX                       WINGAL73
      IF (M .EQ. -1) GO TO 70                                           WINGAL74
C                                                                       WINGAL75
   57 CONTINUE                                                          WINGAL76
      IF (ITIPCOD.EQ.1) CALL TIPSUCT                                    WINGAL77
60    TOTAL=TOTAL-1.                                                    WINGAL78
      GO TO 10                                                          WINGAL79
70    CONTINUE                                                          WINGAL80
C                                                                       WINGAL81
C                                                                       WINGAL82
C                                                                       WINGAL83
80    FORMAT (1H1//10X,I6,93HHORSESHOE VORTICES LAIDOUT, THIS IS MORE THWINGAL84
     1AN THE 400 MAXIMUM. THIS CONFIGURATION IS ABORTED.)               WINGAL85
90    FORMAT (1H1//10X,I6,101H ROWS OF HORSESHOE VORTICES LAIDOUT. THIS WINGAL86
     1IS MORE THAN THE 100 MAXIMUM. THIS CONFIGURATION IS ABORTED)      WINGAL87
100   FORMAT (1H1//10X,8HPLANFORM,I6,4H HAS,I6,74H BREAKPOINTS. THE MAXIWINGAL88
     1MUM DIMENSIONED IS 25. THE CONFIGURATION IS ABORTED.)             WINGAL89
      END                                                               WINGAL90
      SUBROUTINE READIN                                                 READIN 2
C                                                                       READIN 3
C     THIS SUBROUTINE READS IN THE DATA OFF -TAPE81-, AND               READIN 4
C     PRINTS IT TO -OUTPUT-,AND TO -TAPE5-.                             READIN 5
C                                                                       READIN 6
      DIMENSION ICARD(8)                                                READIN 7
      LINE = 0                                                          READIN 8
      WRITE(6,100)                                                      READIN 9
   10 READ(81,200) ICARD                                                READIN10
      IF (EOF(81)) 30, 20                                               READIN11
   20 LINE = LINE + 1                                                   READIN12
      WRITE(6,300) LINE, ICARD                                          READIN13
      IF (LINE .EQ. 1) GO TO 10                                         READIN14
      WRITE(5,200) ICARD                                                READIN15
      GO TO 10                                                          READIN16
C                                                                       READIN17
   30 ENDFILE 5                                                         READIN18
      REWIND 5                                                          READIN19
```

```
C                                                                       READIN20
  100 FORMAT(1H1,30X,11HINPUT  DATA  ,//)                               READIN21
  200 FORMAT(8A10)                                                      READIN22
  300 FORMAT(I10,1H.,1X,8A10)                                           READIN23
      RETURN                                                            READIN24
      END                                                               READIN25
      SUBROUTINE LOADING                                                LOADING2
C                                                                       LOADING3
C      THIS IS A DUMMY ROUTINE USED ONLY TO ENSURE THE                  LOADING4
C      PROPER LOADING OF THE ITEMS NAMED BELOW.                         LOADING5
C                                                                       LOADING6
      CALL STINDX(11,INDEX,1,0)                                         LOADING7
      CALL WRITMS(11,INDEX,1,0)                                         LOADING8
      RETURN                                                            LOADING9
      END                                                               LOADIN10
      SUBROUTINE INFSUB (BOT,FVI,FWI,FUI)                               INFSUB 2
      COMMON /INSUB23/ PSII,APHII,XXX,YYY,ZZZ,SNN,TOLRNC                INFSUB 3
      FC=COS(PSII)                                                      INFSUB 4
      FS=SIN(PSII)                                                      INFSUB 5
      FT=FS/FC                                                          INFSUB 6
C                                                                       INFSUB 7
C                                                                       INFSUB 8
      FPC=COS(APHII)                                                    INFSUB 9
      FPS=SIN(APHII)                                                    INFSUB10
      FPT=FPS/FPC                                                       INFSUB11
      F1=XXX+SNN*FT*FPC                                                 INFSUB12
      F2=YYY+SNN*FPC                                                    INFSUB13
      F3=ZZZ+SNN*FPS                                                    INFSUB14
      F4=XXX-SNN*FT*FPC                                                 INFSUB15
      F5=YYY-SNN*FPC                                                    INFSUB16
      F6=ZZZ-SNN*FPS                                                    INFSUB17
      FFA=(XXX**2+(YYY*FPS)**2+FPC**2*((YYY*FT)**2+(ZZZ/FC)**2-2.*XXX*YYINFSUB18
     1Y*FT)-2.*ZZZ*FPC*(YYY*FPS+XXX*FT*FPS))                            INFSUB19
      FFB=(F1*F1+F2*F2+F3*F3)**.5                                       INFSUB20
      FFC=(F4*F4+F5*F5+F6*F6)**.5                                       INFSUB21
      FFD=F5*F5+F6*F6                                                   INFSUB22
      FFE=F2*F2+F3*F3                                                   INFSUB23
      FFF=(F1*FPC*FT+F2*FPC+F3*FPS)/FFB-(F4*FPC*FT+F5*FPC+F6*FPS)/FFC   INFSUB24
C                                                                       INFSUB25
C                                                                       INFSUB26
C     THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE     INFSUB27
C     CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES              INFSUB28
C                                                                       INFSUB29
C                                                                       INFSUB30
      IF (ABS(FFA).LT.(BOT*15.E-5)**2) GO TO 10                         INFSUB31
      FVONE=(XXX*FPS-ZZZ*FT*FPC)*FFF/FFA                                INFSUB32
      FWONE=(YYY*FT-XXX)*FFF/FFA*FPC                                    INFSUB33
      FUONE=(ZZZ*FPC-YYY*FPS)*FFF/FFA                                   INFSUB34
      GO TO 20                                                          INFSUB35
```

```
10    FUONE=FVONE=FWONE=0.                                              INFSUB36
C                                                                       INFSUB37
20    IF (ABS(FFD).LT.TOLRNC) GO TO 30                                  INFSUB38
      FVTWO=F6*(1.-F4/FFC)/FFD                                          INFSUB39
      FWTWO=-F5*(1.-F4/FFC)/FFD                                         INFSUB40
      GO TO 40                                                          INFSUB41
30    FVTWO=FWTWO=0.                                                    INFSUB42
C                                                                       INFSUB43
40    IF (ABS(FFE).LT.TOLRNC) GO TO 50                                  INFSUB44
      FVTHRE=-F3*(1.-F1/FFB)/FFE                                        INFSUB45
      FWTHRE=F2*(1.-F1/FFB)/FFE                                         INFSUB46
      GO TO 60                                                          INFSUB47
50    FVTHRE=FWTHRE=0.                                                  INFSUB48
C                                                                       INFSUB49
60    FVI=FVONE+FVTWO+FVTHRE                                            INFSUB50
      FWI=FWONE+FWTWO+FWTHRE                                            INFSUB51
      FUI=FUONE                                                         INFSUB52
      RETURN                                                            INFSUB53
      END                                                               INFSUB54
      SUBROUTINE GEOMTRY                                                GEOMTRY2
      DIMENSION NUMBER(4)                                               GEOMTRY3
      DIMENSION XREF(50),YREF(50),SAR(50),A(50),RSAR(50),               GEOMTRY4
     $ X(50), Y(50), SA(4), VBORD(102), SPY(50,4),                      GEOMTRY5
     $ KFX(4),IYL(50,4),IYT(50,4)                                       GEOMTRY6
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                ALL    2
     $     STA(4), TBLSCW(100), YYCP(4),                                ALL    3
     $     Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),        ALL    4
     $     PHI(100), ZH(100), CP(400), STLOIND(4)                       ALL    5
C                                                                       ALL    6
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,    ONETHRE2
     $   ARTRUE, RTCDHT(4), CONFIG(2), NSSWSV(4),                       ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                               ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),      ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                          ONETHRE6
     $  YINNER(4), YOUTER(4)                                            ONETHRE7
C                                                                       ONETHRE8
      INTEGER CONFIG                                                    ONETHRE9
C                                                                       ONETHR10
      COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4),  MAINONE2
     $  XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),       MAINONE3
     $  XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),    MAINONE4
     $  ZZ(25,4), ITIPCOD, ICAMTST                                      MAINONE5
C                                                                       MAINONE6
C                                                                       GEOMTR10
C                                                                       NOTES  2
C             ---  NOTES TO THE USERS  ---                              NOTES  3
C                                                                       NOTES  4
C                                                                       NOTES  5
C       1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING                NOTES  6
```

```
C         EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF              NOTES   7
C         ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE                 NOTES   8
C         CALCULATED FROM CLDES = 100. AND CLDES = 1.0                NOTES   9
C                                                                     NOTES  10
C                                                                     NOTES  11
C       2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOM-   NOTES  12
C          MENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE  NOTES  13
C         MORE MEANINGFUL SIDE EDGE RESULTS                           NOTES  14
C                                                                     NOTES  15
C                                                                     NOTES  16
C                                                                     NOTES  17
C       3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO        NOTES  18
C          0. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE        NOTES  19
C          EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO      NOTES  20
C          1.0 IF THIS LOADING IS TO BE NON-ZERO                      NOTES  21
C                                                                     NOTES  22
C                                                                     NOTES  23
C                                                                     GEOMTR12
      COMMON/CCRRDD/ TSPAN(4), TSPANA, KBIT, CTILDA, XTILDA, DISTALE  CCRRDD 2
C                                                                     CCRRDD 3
      REAL MACH                                                       GEOMTR14
      INTEGER PRTCON                                                  GEOMTR15
      DIMENSION TEMP(8), INDEXES(4,3)                                 GEOMTR16
      DATA INDEXES / 12*1 /                                           GEOMTR17
      DATA NUMBER/5HFIRST,6HSECOND,5HTHIRD,6HFOURTH/                  GEOMTR18
C                                                                     GEOMTR19
C                                                                     GEOMTR20
C       PART ONE - GEOMETRY COMPUTATION                               GEOMTR21
C                                                                     GEOMTR22
C                 SECTION ONE - INPUT OF REFERENCE WING POSITION      GEOMTR23
C                                                                     GEOMTR24
C                                                                     GEOMTR25
      IF(TOTAL .NE. 0.0)GO TO 5                                       GEOMTR26
      DO 4 I=1,4                                                      GEOMTR27
      RTCDHT(I) = 0.0                                                 GEOMTR28
      XT(I) = 0.0                                                     GEOMTR29
      XL(I) = 0.0                                                     GEOMTR30
    4 CONTINUE                                                        GEOMTR31
    5 CONTINUE                                                        GEOMTR32
      YTOL=1.E-10                                                     GEOMTR33
      AZY=1.E+13                                                      GEOMTR34
      PI = 4. * ATAN(1.)                                              GEOMTR35
      PIT = PI / 2.                                                   GEOMTR36
      RAD = 180. / PI                                                 GEOMTR37
      IF (TOTAL.GT.0.) GO TO 70                                       GEOMTR38
C                                                                     GEOMTR39
C                                                                     GEOMTR40
C     SET PLAN EQUAL TO 1. FOR A WING ALONE COMPUTAION - EVEN FOR A   GEOMTR41
C     VARIABLE SWEEP WING                                             GEOMTR42
```

```
C     SET PLAN EQUAL TO 2. FOR A WING - TAIL COMBINATION
C     SET PLAN = 3 OR 4 FOR CANARDS ETC.
C
C     SET TOTAL EQUAL TO THE NUMBER OF SETS
C       OF GROUP TWO DATA PROVIDED
C
      READ(5,885)
     $ PLAN, TOTAL, CREF, SREF, XLOCTN, CTILDA, XTILDA, DISTALE
      IF(EOF(5)) 830,10
10    IPLAN=PLAN
      XTILDA=XTILDA-XLOCTN
C
C
C     SET AAN(IT) EQUAL TO THE MAXIMUM NUMBER OF CURVES REQUIRED TO
C     DEFINE THE PLANFORM PERIMETER OF THE (IT) PLANFORM.
C
C     SET RTCDHT(IT) EQUAL TO THE ROOT CHORD HEIGHT OF THE LIFTING
C     SURFACE (IT),WHOSE PERIMETER POINTS ARE BEING READ IN, WITH
C     RESPECT TO THE WING ROOT CHORD HEIGHT
C
      WRITE (6,860)
      PRTCON = 10H
      DO 60 IT=1,IPLAN
      STLOIND(IT) = 0.0
      READ(5,885)
     $ AAN(IT), XS(IT), YS(IT), RTCDHT(IT), STLOIND(IT)
      N=AAN(IT)
      N1=N+1
      MAK=0
      IF(IPLAN.GT.1) PRTCON = NUMBER(IT)
      WRITE (6,870) PRTCON,N,RTCDHT(IT),XS(IT),YS(IT)
      WRITE (6,990)
      DO 50 I=1,N1
      READ(5,885)
     $ XREG(I,IT), YREG(I,IT), DIH(I,IT), AMCD
C
      IF (AMCD .NE. 2.) AMCD = 1.0
      XREG(I,IT)=XREG(I,IT)-XLOCTN
      MCD(I,IT)=AMCD
      IF (I.EQ.1) GO TO 50
      IF (MAK.NE.0.OR.MCD(I-1,IT).NE.2) GO TO 20
      MAK=I-1
20    IF (ABS(YREG(I-1,IT)-YREG(I,IT)).LT.YTOL) GO TO 30
      AREG(I-1,IT)=(XREG(I-1,IT)-XREG(I,IT))/(YREG(I-1,IT)-YREG(I,IT))
      ASWP=ATAN(AREG(I-1,IT))*RAD
      GO TO 40
30    YREG(I,IT)=YREG(I-1,IT)
      AREG(I-1,IT)=AZY
      ASWP=90.
```

```
40    J=I-1                                                             GEOMTR92
C                                                                       GEOMTR93
C     WRITE PLANFORM PERIMETER POINTS AND ANGLES                        GEOMTR94
C                                                                       GEOMTR95
      WRITE (6,960) J,XREG(J,IT),YREG(J,IT),ASWP,DIH(J,IT),MCD(J,IT)    GEOMTR96
      DIH(J,IT)=TAN(DIH(J,IT)/RAD)                                      GEOMTR97
50    CONTINUE                                                          GEOMTR98
      KFCTS(IT)=MAK                                                     GEOMTR99
      WRITE (6,960) N1,XREG(N1,IT),YREG(N1,IT)                          GEOMT100
60    CONTINUE                                                          GEOMT101
C                                                                       GEOMT102
C                         PART 1 - SECTION 2                            GEOMT103
C         READ GROUP 2 DATA AND COMPUTE DESIRED WING POSITION           GEOMT104
C                                                                       GEOMT105
C                                                                       GEOMT106
C     SET SA(1-IPLAN) EQUAL TO SWEEP ANGLE IN DEGREES                   GEOMT107
C     FOR THE FIRST                                                     GEOMT108
C     CURVE(S) THAT CAN CHANGE SWEEP FOR EACH PLANFORM                  GEOMT109
C                                                                       GEOMT110
C     IF A PARTICULAR VALUE OF CL IS DESIRED AT WHICH THE LOADINGS ARE  GEOMT111
C     TO BE COMPUTED, SET CLDES EQUAL TO THIS VALUE                     GEOMT112
C     SET CLDES EQUAL TO 11. FOR A DRAG POLAR AT CL VALUES OF-.1 TO  1.0GEOMT113
C                                                                       GEOMT114
C     IF PTEST IS SET EQUAL TO ONE THE PROGRAM WILL COMPUTE  CLP        GEOMT115
C     IF QTEST IS SET EQUAL TO ONE THE PROGRAM WILL COMPUTE  CMQ AND CLQGEOMT116
C     DO NOT SET BOTH PTEST AND QTEST TO ONE FOR A SINGLE CONFIGURATION GEOMT117
C                                                                       GEOMT118
C     SET TWIST(1-IPLAN) EQUAL TO 0. FOR A FLAT PLANFORM                GEOMT119
C     AND TO 1.                                                         GEOMT120
C     FOR A PLANFORM THAT HAS TWIST AND/OR CAMBER                       GEOMT121
C                                                                       GEOMT122
C     SET ATPCOD TO ONE IF THE CONTRIBUTIONS TO LIFT,DRAG AND MOMENT    GEOMT123
C     FROM SEPERATED FLOW AROUND THE LEADING AND/OR SIDE EDGES IS       GEOMT124
C     DESIRED.   OTHERWISE SET ATPCOD TO ZERO.                          GEOMT125
C                                                                       GEOMT126
C                                                                       GEOMT127
   70 READ(5,950) CONFIG(1), CONFIG(2), SCW, VIC, MACH, CLDES,          GEOMT128
     $  (SA(I),I=1,4),(TWIST(J),J=1,4),PTEST,                           GEOMT129
     $   QTEST,ATPCOD                                                   GEOMT130
C                                                                       GEOMT131
      IF (EOF(5)) 830, 71                                               GEOMT132
   71 IF (VIC .GE. 10.) GO TO 72                                        GEOMT133
      WRITE(6,1070)                                                     GEOMT134
      M = -1                                                            GEOMT135
      GO TO 2000                                                        GEOMT136
   72 IF( CLDES .NE. 100. .OR. ATPCOD .NE. 1.) GO TO 73                 GEOMT137
      IF (CLDES .EQ. 100. .AND. ATPCOD .EQ. 1. .AND.                    GEOMT138
     $    SCW .GE. 2.) GO TO 73                                         GEOMT139
      WRITE(6,1080)                                                     GEOMT140
```

```
      M = -1                                                  GEOMT141
      GO TO 2000                                              GEOMT142
   73 ICAMTST = 0                                             GEOMT143
      DO 74 IT = 1,4                                          GEOMT144
        IF(TWIST(IT) .NE. 0.) ICAMTST = 1                     GEOMT145
   74 CONTINUE                                                GEOMT146
      IF(ATPCOD .EQ. 1.) ICAMTST = ICAMTST + 1                GEOMT147
      IF(CLDES .NE. 100.) ICAMTST = ICAMTST + 1               GEOMT148
      IF(ICAMTST .NE. 3) GO TO 75                             GEOMT149
      WRITE(6,1060)                                           GEOMT150
      GO TO 2000                                              GEOMT151
   75 ITIPCOD = ATPCOD                                        GEOMT152
      IF (ATPCOD .NE. 2.) GO TO 90                            GEOMT153
      IF (CLDES .NE. 11. .OR. CLDES .NE. 100.) GO TO 90       GEOMT154
      M = -1                                                  GEOMT155
      WRITE(6,1100) CLDES                                     GEOMT156
      GO TO 2000                                              GEOMT157
C                                                             GEOMT158
   90 IF (ITIPCOD .NE. 1) GO TO 110                           GEOMT159
C                                                             GEOMT160
C            READ IN THE LIMITS OF INTEGRATION ON             GEOMT161
C            THE SIDE EDGES, AND ON THE LEADING/TRAILING      GEOMT162
C            EDGES                                            GEOMT163
C            (IF ATPCOD = 1)                                  GEOMT164
C                                                             GEOMT165
      READ(5,885) (YINNER(IT),YOUTER(IT),IT=1,IPLAN)          GEOMT166
      READ(5,885) (XL(IT), XT(IT), IT=1,IPLAN)                GEOMT167
      DO 100 IT= 1,IPLAN                                      GEOMT168
        XL(IT) = XL(IT) - XLOCTN                              GEOMT169
        XT(IT) = XT(IT) - XLOCTN                              GEOMT170
100   CONTINUE                                                GEOMT171
      IF (SCW .NE. 0.) GO TO 110                              GEOMT172
      I = 1                                                   GEOMT173
      DO 107 ITT = 1,IPLAN                                    GEOMT174
        IF (XL(ITT) .NE. XT(ITT)) I = I + 1                   GEOMT175
  107 CONTINUE                                                GEOMT176
      IF (I .EQ. 0) GO TO 110                                 GEOMT177
      M = -1                                                  GEOMT178
      WRITE(6,1110)                                           GEOMT179
      GO TO 2000                                              GEOMT180
C                                                             GEOMT181
110   CONTINUE                                                GEOMT182
      WRITE (6,890) CONFIG                                    GEOMT183
      IF (PTEST .EQ. 1. .AND. QTEST .EQ. 1.) GO TO 850        GEOMT184
      IF (SCW.EQ.0.) GO TO 140                                GEOMT185
      DO 125 I = 1,100                                        GEOMT186
        TBLSCW(I) = SCW                                       GEOMT187
  125 CONTINUE                                                GEOMT188
      GO TO 150                                               GEOMT189
```

```
C                                                                       GEOMT190
C     READ IN THE -STA- AND -TBLSCW- VALUES                             GEOMT191
  140 ISTART = 0                                                        GEOMT192
      IEND = 0                                                          GEOMT193
      NSTA = 0                                                          GEOMT194
      DO 145 ITT = 1,IPLAN                                              GEOMT195
        READ(5,880) STA(ITT)                                            GEOMT196
        NSTA = NSTA + STA(ITT)                                          GEOMT197
        ISTART = IEND + 1                                               GEOMT198
        IEND = NSTA                                                     GEOMT199
        READ(5,890) (TBLSCW(IKK),IKK=ISTART,IEND)                       GEOMT200
  145 CONTINUE                                                          GEOMT201
C                                                                       GEOMT202
C                                                                       GEOMT203
150   DO 410 IT=1,IPLAN                                                 GEOMT204
      N=AAN(IT)                                                         GEOMT205
      N1=N+1                                                            GEOMT206
      DO 160 I=1,N                                                      GEOMT207
      XREF(I)=XREG(I,IT)                                                GEOMT208
      YREF(I)=YREG(I,IT)                                                GEOMT209
      A(I)=AREG(I,IT)                                                   GEOMT210
      RSAR(I)=ATAN(A(I))                                                GEOMT211
      IF (A(I).EQ.AZY) RSAR(I)=PIT                                      GEOMT212
160   CONTINUE                                                          GEOMT213
      XREF(N1)=XREG(N1,IT)                                              GEOMT214
      YREF(N1)=YREG(N1,IT)                                              GEOMT215
      IF (KFCTS(IT).GT.0) GO TO 170                                     GEOMT216
      K=1                                                               GEOMT217
      SA(IT)=RSAR(1)*RAD                                                GEOMT218
      GO TO 180                                                         GEOMT219
170   K=KFCTS(IT)                                                       GEOMT220
180   WRITE (6,920) K,SA(IT),IT                                         GEOMT221
      SB=SA(IT)/RAD                                                     GEOMT222
      IF (ABS(SB-RSAR(K)).GT.(.1/RAD)) GO TO 210                        GEOMT223
C     REFERENCE PLANFORM COORDINATES ARE STORED UNCHANGED FOR WINGS     GEOMT224
C             WITHOUT CHANGE IN SWEEP                                   GEOMT225
      DO 200 I=1,N                                                      GEOMT226
      X(I)=XREF(I)                                                      GEOMT227
      Y(I)=YREF(I)                                                      GEOMT228
      IF (RSAR(I).EQ.PIT) GO TO 190                                     GEOMT229
      A(I)=TAN(RSAR(I))                                                 GEOMT230
      GO TO 200                                                         GEOMT231
190   A(I)=AZY                                                          GEOMT232
200   SAR(I)=RSAR(I)                                                    GEOMT233
      X(N1)=XREF(N1)                                                    GEOMT234
      Y(N1)=YREF(N1)                                                    GEOMT235
      GO TO 390                                                         GEOMT236
C                                                                       GEOMT237
C     CHANGES IN WING SWEEP ARE MADE HERE                               GEOMT238
```

```
C                                                                       GEOMT239
210   IF (MCD(K,IT).NE.2) GO TO 840                                     GEOMT240
      KA=K-1                                                            GEOMT241
      DO 220 I=1,KA                                                     GEOMT242
      X(I)=XREF(I)                                                      GEOMT243
      Y(I)=YREF(I)                                                      GEOMT244
220   SAR(I)=RSAR(I)                                                    GEOMT245
C     DETERMINE LEADING EDGE INTERSECTION BETWEEN FIXED AND VARIABLE    GEOMT246
C            SWEEP WING SECTIONS                                        GEOMT247
      SAR(K)=SB                                                         GEOMT248
      A(K)=TAN(SB)                                                      GEOMT249
      SAI=SB-RSAR(K)                                                    GEOMT250
      X(K+1)=XS(IT)+(XREF(K+1)-XS(IT))*COS(SAI)+(YREF(K+1)-YS(IT))*SIN(SGEOMT251
     1AI)                                                               GEOMT252
      Y(K+1)=YS(IT)+(YREF(K+1)-YS(IT))*COS(SAI)-(XREF(K+1)-XS(IT))*SIN(SGEOMT253
     1AI)                                                               GEOMT254
      IF (ABS(SB-SAR(K-1)).LT.(.1/RAD)) GO TO 230                       GEOMT255
      Y(K)=X(K+1)-X(K-1)-A(K)*Y(K+1)+A(K-1)*Y(K-1)                      GEOMT256
      Y(K)=Y(K)/(A(K-1)-A(K))                                           GEOMT257
      X(K)=A(K)*X(K-1)-A(K-1)*X(K+1)+A(K-1)*A(K)*(Y(K+1)-Y(K-1))        GEOMT258
      X(K)=X(K)/(A(K)-A(K-1))                                           GEOMT259
      GO TO 240                                                         GEOMT260
C     ELIMINATE EXTRANEOUS BREAKPOINTS                                  GEOMT261
230   X(K)=XREF(K-1)                                                    GEOMT262
      Y(K)=YREF(K-1)                                                    GEOMT263
      SAR(K)=SAR(K-1)                                                   GEOMT264
240   K=K+1                                                             GEOMT265
C     SWEEP THE BREAKPOINTS ON THE VARIABLE SWEEP PANEL                 GEOMT266
C        (IT ALSO KEEPS SWEEP ANGLES IN FIRST OR FOURTH QUADRANTS)      GEOMT267
250   K=K+1                                                             GEOMT268
      SAR(K-1)=SAI+RSAR(K-1)                                            GEOMT269
260   IF (SAR(K-1).LE.PIT) GO TO 270                                    GEOMT270
      SAR(K-1)=SAR(K-1)-3.1415927                                       GEOMT271
      GO TO 260                                                         GEOMT272
270   IF (SAR(K-1).GE.(-PIT)) GO TO 280                                 GEOMT273
      SAR(K-1)=SAR(K-1)+3.1415927                                       GEOMT274
      GO TO 270                                                         GEOMT275
280   IF ((SAR(K-1)).LT..0) GO TO 290                                   GEOMT276
      IF (SAR(K-1)-PIT) 320,300,300                                     GEOMT277
290   IF (SAR(K-1)+PIT) 310,310,320                                     GEOMT278
300   A(K-1)=AZY                                                        GEOMT279
      GO TO 330                                                         GEOMT280
310   A(K-1)=-AZY                                                       GEOMT281
      GO TO 330                                                         GEOMT282
320   A(K-1)=TAN(SAR(K-1))                                              GEOMT283
330   KK=MCD(K,IT)                                                      GEOMT284
      IF(KK .EQ. 1) GO TO 350                                           GEOMT285
340   Y(K)=YS(IT)+(YREF(K)-YS(IT))*COS(SAI)-(XREF(K)-XS(IT))*SIN(SAI)   GEOMT286
      X(K)=XS(IT)+(XREF(K)-XS(IT))*COS(SAI)+(YREF(K)-YS(IT))*SIN(SAI)   GEOMT267
```

```
      GO TO 250                                                         GEOMT288
C     DETERMINE THE TRAILING EDGE INTERSECTION                          GEOMT289
C        BETWEEN FIXED AND VARIABLE SWEEP WING SECTIONS                 GEOMT290
350   IF (ABS(RSAR(K)-SAP(K-1)).LT.(.1/RAD)) GO TO 360                  GEOMT291
      Y(K)=XREF(K+1)-X(K-1)-A(K)*YREF(K+1)+A(K-1)*Y(K-1)                GEOMT292
      Y(K)=Y(K)/(A(K-1)-A(K))                                           GEOMT293
      X(K)=A(K)*X(K-1)-A(K-1)*XREF(K+1)+A(K-1)*A(K)*(YREF(K+1)-Y(K-1))  GEOMT294
      X(K)=X(K)/(A(K)-A(K-1))                                           GEOMT295
      GO TO 370                                                         GEOMT296
360   X(K)=XREF(K+1)                                                    GEOMT297
      Y(K)=YREF(K+1)                                                    GEOMT298
370   K=K+1                                                             GEOMT299
C     STORE REFERENCE PLANFORM COORDINATES ON INBOARD FIXED TRAILING    GEOMT300
C     EDGE                                                              GEOMT301
      DO 380 I=K,N1                                                     GEOMT302
      X(I)=XREF(I)                                                      GEOMT303
      Y(I)=YREF(I)                                                      GEOMT304
380   SAR(I-1)=RSAR(I-1)                                                GEOMT305
390   DO 400 I=1,N                                                      GEOMT306
      XX(I,IT)=X(I)                                                     GEOMT307
      YY(I,IT)=Y(I)                                                     GEOMT308
      MMCD(I,IT)=MCD(I,IT)                                              GEOMT309
      TTWD(I,IT)=DIH(I,IT)                                              GEOMT310
400   AS(I,IT)=A(I)                                                     GEOMT311
      XX(N1,IT)=X(N1)                                                   GEOMT312
      YY(N1,IT)=Y(N1)                                                   GEOMT313
      AN(IT)=AAN(IT)                                                    GEOMT314
410   CONTINUE                                                          GEOMT315
C                                                                       GEOMT316
C       LINE UP BREAKPOINTS AMONG PLANFORMS                             GEOMT317
C                                                                       GEOMT318
      BOTSV(1) = 0.0                                                    GEOMT319
      BOTSV(2) = 0.0                                                    GEOMT320
      BOTSV(3) = 0.0                                                    GEOMT321
      BOTSV(4) = 0.0                                                    GEOMT322
      WRITE (6,980)                                                     GEOMT323
      DO 530 IT=1,IPLAN                                                 GEOMT324
      NIT=AN(IT)+1                                                      GEOMT325
      DO 470 ITT=1,IPLAN                                                GEOMT326
      IF (ITT.EQ.IT) GO TO 470                                          GEOMT327
      NITT=AN(ITT)+1                                                    GEOMT328
      DO 460 I=1,NITT                                                   GEOMT329
      JPSV=0                                                            GEOMT330
      DO 420 JP=1,NIT                                                   GEOMT331
      IF (YY(JP,IT).EQ.YY(I,ITT)) GO TO 460                             GEOMT332
420   CONTINUE                                                          GEOMT333
      DO 430 JP=1,NIT                                                   GEOMT334
      IF (YY(JP,IT).LT.YY(I,ITT)) GO TO 440                             GEOMT335
430   CONTINUE                                                          GEOMT336
```

```
      GO TO 460                                                         GEOMT337
440   IF(JP .EQ. 1) GO TO 460                                           GEOMT338
      JPSV = JP                                                         GEOMT339
      IND=NIT-(JPSV-1)                                                  GEOMT340
      DO 450 JP=1,IND                                                   GEOMT341
      K2=NIT-JP+2                                                       GEOMT342
      K1=NIT-JP+1                                                       GEOMT343
      XX(K2,IT)=XX(K1,IT)                                               GEOMT344
      YY(K2,IT)=YY(K1,IT)                                               GEOMT345
      MMCD(K2,IT)=MMCD(K1,IT)                                           GEOMT346
      AS(K2,IT)=AS(K1,IT)                                               GEOMT347
450   TTWD(K2,IT)=TTWD(K1,IT)                                           GEOMT348
      YY(JPSV,IT)=YY(I,ITT)                                             GEOMT349
      AS(JPSV,IT)=AS(JPSV-1,IT)                                         GEOMT350
      TTWD(JPSV,IT)=TTWD(JPSV-1,IT)                                     GEOMT351
      XX(JPSV,IT)=(YY(JPSV,IT)-YY(JPSV-1,IT))*AS(JPSV-1,IT)+XX(JPSV-1,ITGEOMT352
     1)                                                                 GEOMT353
      MMCD(JPSV,IT)=MMCD(JPSV-1,IT)                                     GEOMT354
      AN(IT)=AN(IT)+1.                                                  GEOMT355
      NIT=NIT+1                                                         GEOMT356
460   CONTINUE                                                          GEOMT357
470   CONTINUE                                                          GEOMT358
C                                                                       GEOMT359
C     SEQUENCE WING COORDINATES FROM TIP TO ROOT                        GEOMT360
C                                                                       GEOMT361
      N1=AN(IT)+1.                                                      GEOMT362
      DO 480 I=1,N1                                                     GEOMT363
480   Q(I)=YY(I,IT)                                                     GEOMT364
      DO 520 J=1,N1                                                     GEOMT365
      HIGH=1.                                                           GEOMT366
      DO 490 I=1,N1                                                     GEOMT367
      IF ((Q(I)-HIGH).GE.0.) GO TO 490                                  GEOMT368
      HIGH=Q(I)                                                         GEOMT369
      IH=I                                                              GEOMT370
490   CONTINUE                                                          GEOMT371
      IF (J.NE.1) GO TO 500                                             GEOMT372
      BOTSV(IT)=HIGH                                                    GEOMT373
      KFX(IT)=IH                                                        GEOMT374
500   Q(IH)=1.                                                          GEOMT375
      SPY(J,IT)=HIGH                                                    GEOMT376
      IF (IH.GT.KFX(IT)) GO TO 510                                      GEOMT377
      IYL(J,IT)=1                                                       GEOMT378
      IYT(J,IT)=0                                                       GEOMT379
      GO TO 520                                                         GEOMT380
510   IYL(J,IT)=0                                                       GEOMT381
      IYT(J,IT)=1                                                       GEOMT382
520   CONTINUE                                                          GEOMT383
530   CONTINUE                                                          GEOMT384
C                                                                       GEOMT385
```

```
C     SELECT MAXIMUM B/2 AS THE WING SEMISPAN.   IF BOTH FIRST AND        GEOMT386
C     SECOND PLANFORMS HAVE SAME SEMISPAN THEN THE SECOND PLANFORM IS     GEOMT387
C     TAKEN TO BE THE WING.                                               GEOMT388
C                                                                         GEOMT389
      KBOT=1                                                              GEOMT390
      DO 535 IT = 1,IPLAN                                                 GEOMT391
      IF(ABS(BOTSV(IT)).GE.ABS(BOTSV(KBOT))) KBOT = IT                    GEOMT392
  535 CONTINUE                                                            GEOMT393
      BOT=BOTSV(KBOT)                                                     GEOMT394
C                                                                         GEOMT395
C     COMPUTE NOMINAL HORSESHOE VORTEX WIDTH ALONG WING SURFACE           GEOMT396
C                                                                         GEOMT397
      DO 570 IT = 1,IPLAN                                                 GEOMT398
      TSPAN(IT) = 0.0                                                     GEOMT399
      ISAVE = KFX(IT) - 1                                                 GEOMT400
      I = KFX(IT) - 2                                                     GEOMT401
  540 IF(I.EQ.0) GOTO 550                                                 GEOMT402
      IF(TTWD(I,IT).EQ.TTWD(ISAVE,IT)) GOTO 560                           GEOMT403
  550 CTWD = COS(ATAN(TTWD(ISAVE,IT)))                                    GEOMT404
      TLGTH = (YY(ISAVE+1,IT)-YY(I+1,IT))/CTWD                            GEOMT405
      TSPAN(IT) = TSPAN(IT) + TLGTH                                       GEOMT406
      IF (I.EQ.0) GO TO 570                                               GEOMT407
      ISAVE=I                                                             GEOMT408
560   I=I-1                                                               GEOMT409
      GO TO 540                                                           GEOMT410
  570 CONTINUE                                                            GEOMT411
      VI = TSPAN(KBOT) / VIC                                              GEOMT412
      VSTOL=VI/2                                                          GEOMT413
C                                                                         GEOMT414
C     ELIMINATE PLANFORM BREAKPOINTS WHICH ARE WITHIN (B/2)/2000 UNITS    GEOMT415
C     LATERALLY                                                           GEOMT416
C                                                                         GEOMT417
      DO 630 IT=1,IPLAN                                                   GEOMT418
      N=AN(IT)                                                            GEOMT419
      N1=N+1                                                              GEOMT420
      DO 630 J=1,N                                                        GEOMT421
      AA=ABS(SPY(J,IT)-SPY(J+1,IT))                                       GEOMT422
      IF(AA.EQ.0..OR.AA.GT.ABS(TSPAN(KBOT)/2000.)) GOTO 630               GEOMT423
      IF (AA.GT.YTOL) WRITE (6,1010) SPY(J+1,IT),SPY(J,IT)                GEOMT424
      DO 620 I=1,N1                                                       GEOMT425
      IF (YY(I,IT).NE.SPY(J+1,IT)) GO TO 620                              GEOMT426
      YY(I,IT)=SPY(J,IT)                                                  GEOMT427
620   CONTINUE                                                            GEOMT428
      SPY(J+1,IT)=SPY(J,IT)                                               GEOMT429
630   CONTINUE                                                            GEOMT430
C                                                                         GEOMT431
C     COMPUTE Z COORDINATES                                               GEOMT432
C                                                                         GEOMT433
      DO 670 IT=1,IPLAN                                                   GEOMT434
```

```
      JM=N1=AN(IT)+1.                                                    GEOMT435
      DO 640 JZ=1,N1                                                     GEOMT436
640   ZZ(JZ,IT)=RTCDHT(IT)                                               GEOMT437
      JZ=1                                                               GEOMT438
650   JZ=JZ+1                                                            GEOMT439
      IF (JZ.GT.KFX(IT)) GO TO 660                                       GEOMT440
      ZZ(JZ,IT)=ZZ(JZ-1,IT)+(YY(JZ,IT)-YY(JZ-1,IT))*TTWD(JZ-1,IT)        GEOMT441
      GO TO 650                                                          GEOMT442
660   JM=JM-1                                                            GEOMT443
      IF (JM.EQ.KFX(IT)) GO TO 670                                       GEOMT444
      ZZ(JM,IT)=ZZ(JM+1,IT)+(YY(JM,IT)-YY(JM+1,IT))*TTWD(JM,IT)          GEOMT445
      GO TO 660                                                          GEOMT446
670   CONTINUE                                                           GEOMT447
C                                                                        GEOMT448
C     WRITE PLANFORM PERIMETER POINTS ACTUALLY USED IN THE COMPUTATIONS GEOMT449
C                                                                        GEOMT450
      WRITE (6,900)                                                      GEOMT451
      DO 690 IT=1,IPLAN                                                  GEOMT452
      N=AN(IT)                                                           GEOMT453
      N1=N+1                                                             GEOMT454
      IF(IPLAN.GT.1) WRITE(6,1000) NUMBER(IT)                            GEOMT455
      DO 680 KK=1,N                                                      GEOMT456
      TOUT=ATAN(TTWD(KK,IT))*RAD                                         GEOMT457
      AOUT=ATAN(AS(KK,IT))*RAD                                           GEOMT458
      IF (AS(KK,IT).EQ.AZY) AOUT=90.                                     GEOMT459
      WRITE (6,910) KK,XX(KK,IT),YY(KK,IT),ZZ(KK,IT),AOUT,TOUT,MMCD(KK,IGEOMT460
     1T)                                                                 GEOMT461
680   CONTINUE                                                           GEOMT462
      WRITE (6,910) N1,XX(N1,IT),YY(N1,IT),ZZ(N1,IT)                     GEOMT463
690   CONTINUE                                                           GEOMT464
C                                                                        GEOMT465
C     PART ONE - SECTION THREE - LAY OUT YAWED HORSESHOE VORTICES        GEOMT466
C                                                                        GEOMT467
      STRUE=0.                                                           GEOMT468
      DO 695 IT = 1,IPLAN                                                GEOMT469
      NSSWSV(IT) = 0                                                     GEOMT470
      MSV(IT) = 0                                                        GEOMT471
  695 CONTINUE                                                           GEOMT472
      MSTOT = 0                                                          GEOMT473
      NSWTOT = 0                                                         GEOMT474
      DO 780 IT=1,IPLAN                                                  GEOMT475
      N1=AN(IT)+1.                                                       GEOMT476
      I=0                                                                GEOMT477
      J=1                                                                GEOMT478
      YIN=BOTSV(IT)                                                      GEOMT479
      ILE=ITE=KFX(IT)                                                    GEOMT480
C     DETERMINE SPANWISE BORDERS OF HORSESHOE VORTICES                   GEOMT481
700   IXL=IXT=0                                                          GEOMT482
      I=I+1                                                              GEOMT483
```

```
      CPHI = COS(ATAN(TTWD(ILE,IT)))                                   GEOMT484
      IF (YIN.GE.(SPY(J,IT)+VSTOL*CPHI)) GO TO 710                     GEOMT485
C     BORDER IS WITHIN VORTEX SPACING TOLERANCE (VSTOL) OF BREAKPOINT  GEOMT486
C     THEREFORE USE THE NEXT BREAKPOINT INBOARD FOR THE BORDER         GEOMT487
      VBORD(I)=YIN                                                     GEOMT488
      GO TO 740                                                        GEOMT489
C     USE NOMINAL VORTEX SPACING TO DETERMINE THE BORDER               GEOMT490
710   VBORD(I)=SPY(J,IT)                                               GEOMT491
C     COMPUTE SUBSCRIPTS ILE AND ITE TO INDICATE WHICH                 GEOMT492
C     BREAKPOINTS ARE ADJACENT AND WHETHER THEY ARE ON THE WING LEADING GEOMT493
C       EDGE OR THE TRAILING EDGE                                      GEOMT494
720   IF (J.GE.N1) GO TO 730                                           GEOMT495
      IF (SPY(J,IT).NE.SPY(J+1,IT)) GO TO 730                          GEOMT496
      IXL=IXL+IYL(J,IT)                                                GEOMT497
      IXT=IXT+IYT(J,IT)                                                GEOMT498
      J=J+1                                                            GEOMT499
      GO TO 720                                                        GEOMT500
730   YIN=SPY(J,IT)                                                    GEOMT501
      IXL=IXL+IYL(J,IT)                                                GEOMT502
      IXT=IXT+IYT(J,IT)                                                GEOMT503
      J=J+1                                                            GEOMT504
740   CONTINUE                                                         GEOMT505
      IPHI=ILE-IXL                                                     GEOMT506
      IF (J.GE.N1) IPHI=1                                              GEOMT507
      YIN=YIN-VI*COS(ATAN(TTWD(IPHI,IT)))                              GEOMT508
      IF (I.NE.1) GO TO 760                                            GEOMT509
750   ILE=ILE-IXL                                                      GEOMT510
      ITE=ITE+IXT                                                      GEOMT511
      GO TO 700                                                        GEOMT512
C     COMPUTE COORDINATES FOR CHORDWISE ROW OF HORSESHOE VORTICES      GEOMT513
760   YQ=(VBORD(I-1)+VBORD(I))/2.                                      GEOMT514
      HW=(VBORD(I)-VBORD(I-1))/2.                                      GEOMT515
      IM1 = I - 1 + NSWTOT                                             GEOMT516
      ZH(IM1)=ZZ(ILE,IT)+(YQ-YY(ILE,IT))*TTWD(ILE,IT)                  GEOMT517
      PHI(IM1)=TTWD(ILE,IT)                                            GEOMT518
      SSWWA(IM1)=AS(ILE,IT)                                            GEOMT519
      XLE=XX(ILE,IT)+AS(ILE,IT)*(YQ-YY(ILE,IT))                        GEOMT520
      XTE=XX(ITE,IT)+AS(ITE,IT)*(YQ-YY(ITE,IT))                        GEOMT521
      XLOCAL=(XLE-XTE)/TBLSCW(IM1)                                     GEOMT522
C                                                                      GEOMT523
C     COMPUTE WING AREA PROJECTED TO THE X - Y PLANE                   GEOMT524
C                                                                      GEOMT525
      STRUE=STRUE+XLOCAL*TBLSCW(IM1)*(HW*2.)*2.                        GEOMT526
C                                                                      GEOMT527
      NSCW=TBLSCW(IM1)                                                 GEOMT528
      DO 770 JCW=1,NSCW                                                GEOMT529
      AJCW=JCW-1                                                       GEOMT530
      XLEL=XLE-AJCW*XLOCAL                                             GEOMT531
      NTS = JCW + MSTOT                                                GEOMT532
```

```
      PN(NTS)=XLEL-.25*XLOCAL                                           GEOMT533
      PV(NTS)=XLEL-.75*XLOCAL                                           GEOMT534
      PSI(NTS)=((XLE-PN(NTS))*AS(ITE,IT)+(PN(NTS)-XTE)*AS(ILE,IT))/(XLE-GEOMT535
     1XTE)                                                              GEOMT536
      S(NTS)=HW/CPHI                                                    GEOMT537
      Q(NTS)=YQ                                                         GEOMT538
  770 CONTINUE                                                          GEOMT539
      MSV(IT)=MSV(IT)+NSCW                                              GEOMT540
      MSTOT = MSTOT + NSCW                                              GEOMT541
C                                                                       GEOMT542
C     TEST TO DETERMINE WHEN WING ROOT IS REACHED                       GEOMT543
      IF (VBORD(I).LT.YREG(1,IT)) GO TO 750                             GEOMT544
C                                                                       GEOMT545
      NSSWSV(IT)=I-1                                                    GEOMT546
      NSWTOT = NSWTOT + NSSWSV(IT)                                      GEOMT547
  780 CONTINUE                                                          GEOMT548
      M = 0                                                             GEOMT549
      DO 781 IT = 1,IPLAN                                               GEOMT550
      M = M + MSV(IT)                                                   GEOMT551
  781 CONTINUE                                                          GEOMT552
C                                                                       GEOMT553
C     COMPUTE ASPECT RAT1O  AND  AVERAGE CHORD                          GEOMT554
C                                                                       GEOMT555
      BOT=-BOT                                                          GEOMT556
      AR=4.*BOT*BOT/SREF                                                GEOMT557
      ARTRUE=4.*BOT*BOT/STRUE                                           GEOMT558
      CAVE=STRUE/(2.*BOT)                                               GEOMT559
      BETA=(1.-MACH*MACH)**.5                                           GEOMT560
C                                                                       GEOMT561
C     READ IN THE -ALP- VALUES. IF TWIST(ITT) = 0.,                     GEOMT562
C     THEN BYPASS THE READ AND SIMPLY SET THEM TO 0.                    GEOMT563
C                                                                       GEOMT564
      DEGRAD = 1. / RAD                                                 GEOMT565
      ISTART = 0                                                        GEOMT566
      IEND = 0                                                          GEOMT567
      INDXTBL = 0                                                       GEOMT568
      DO 810 ITT = 1,IPLAN                                              GEOMT569
        JRANGE = NSSWSV(ITT)                                            GEOMT570
        DO 805 J = 1,JRANGE                                             GEOMT571
          INDXTBL = INDXTBL + 1                                         GEOMT572
          NUMOALP = TBLSCW(INDXTBL)                                     GEOMT573
          ISTART = IEND + 1                                             GEOMT574
          IEND = IEND + NUMOALP                                         GEOMT575
          IF (TWIST(ITT) .EQ. 0.) GO TO 795                             GEOMT576
          READ(5,985) (ALP(ITOE),ITOE=ISTART,IEND)                      GEOMT577
          IF (TWIST(ITT) .EQ. 1.) GO TO 805                             GEOMT578
          DO 790 ITOE = ISTART,IEND                                     GEOMT579
            ALP(ITOE) = ALP(ITOE) * DEGRAD                              GEOMT580
  790     CONTINUE                                                      GEOMT581
```

```
      GO TO 805                                               GEOMT582
  795 DO 800 ITOE = ISTART,IEND                               GEOMT583
        ALP(ITOE) = 0.0                                       GEOMT584
  800   CONTINUE                                              GEOMT585
  805  CONTINUE                                               GEOMT586
  810 CONTINUE                                                GEOMT587
C                                                             GEOMT588
C                                                             GEOMT589
C                                                             GEOMT590
      WRITE (6,1040) M                                        GEOMT591
      WRITE (6,1050) (IT,MSV(IT),NSSWSV(IT),IT=1,IPLAN)       GEOMT592
      NUMHORV = SCW                                           GEOMT593
      IF ( SCW .NE. 0.) GO TO 815                             GEOMT594
C                                                             GEOMT595
C     SCW = 0, SO THE USER INPUT THE VALUES OF -TBLSCW-.      GEOMT596
C     CHECK THE NUMBER OF INPUT STATIONS, -STA-, AGAINST      GEOMT597
C     THE NUMBER COMPUTED IN -NSSWSV-                         GEOMT598
C                                                             GEOMT599
      WRITE(6,1030) (TBLSCW(I),I=1,NSTA)                      GEOMT600
      I = 0                                                   GEOMT601
      DO 813 ITT = 1,IPLAN                                    GEOMT602
        J = STA(ITT)                                          GEOMT603
        IF (J .NE. NSSWSV(ITT)) I = 1                         GEOMT604
  813 CONTINUE                                                GEOMT605
      IF (I .EQ. 0) GO TO 817                                 GEOMT606
      M = -1                                                  GEOMT607
      WRITE(6,1090)                                           GEOMT608
      GO TO 2000                                              GEOMT609
C                                                             GEOMT610
  815 WRITE(6,1020) NUMHORV                                   GEOMT611
  817 CONTINUE                                                GEOMT612
C                                                             GEOMT613
C               PLOT PLANFORM CONFIGURATIONS ON THE           GEOMT614
C               LINE PRINTER                                  GEOMT615
C                                                             GEOMT616
      CALL PLANPLT(IPLAN,XX,YY,AN,0,0)                        GEOMT617
C                                                             GEOMT618
C     APPLY PRANDTL-GLAUERT CORRECTION                        GEOMT619
C                                                             GEOMT620
      DO 820 NV=1,M                                           GEOMT621
      PSI(NV)=ATAN(PSI(NV)/BETA)                              GEOMT622
      PN(NV)=PN(NV)/BETA                                      GEOMT623
  820 PV(NV)=PV(NV)/BETA                                      GEOMT624
      GO TO 2000                                              GEOMT625
  830 ICODEOF=1                                               GEOMT626
      WRITE (6,930) CONFIG                                    GEOMT627
      GO TO 2000                                              GEOMT628
  840 ICODEOF=2                                               GEOMT629
      WRITE (6,940) K,IT                                      GEOMT630
```

```
      GO TO 2000
850   ICODEOF=3
      WRITE (6,970) PTEST,QTEST
      GO TO 2000
C
C
C
860   FORMAT (1H1//63X,13HGEOMETRY DATA)
870   FORMAT (///45X,A10,22HREFERENCE PLANFORM HAS,I3,7H CURVES//12X,19H
     1ROOT CHORD HEIGHT =,F12.5,4X,29HVARIABLE SWEEP PIVOT POSITION,4X,6
     2HX(S) =,F12.5,5X,6HY(S) =,F12.5//46X,40HBREAK POINTS FOR THE REFER
     3ENCE PLANFORM /)
  880 FORMAT(16F5.1)
  885 FORMAT(8F10.6)
  890 FORMAT(1H1,//,47X,16HCONFIGURATION : ,2A10)
900   FORMAT (22X,5HPOINT,6X,1HX,11X,1HY,11X,1HZ,10X,5HSWEEP,7X,8HDIHEDR
     1AL,4X,4HMOVE/68X,5HANGLE,8X,5HANGLE,6X,4HCODE/)
910   FORMAT (20X,I5,3F12.5,2F14.5,I6)
920   FORMAT (/40X,5HCURVE,I3,9H IS SWEPT,F12.5,20H DEGREES ON PLANFORM,
     1I3)
  930 FORMAT(1H1,//,41X,
     $ 43HEND OF FILE ENCOUNTERED AFTER CONFIGURATION ,1X,2A10)
940   FORMAT (1H1///18X,45HTHE FIRST VARIABLE SWEEP CURVE SPECIFIED (K =
     1,I3,44H ) DOES NOT HAVE AN M CODE OF 2 FOR PLANFORM,I4)
  950 FORMAT(2A10,8F5.2,7F2.0)
960   FORMAT (26X,I5,2F12.5,2F16.5,4X,I4)
970   FORMAT (1H1//30X,38HERROR - PROGRAM CANNOT PROCESS PTEST =,F5.1,12
     1H AND QTEST =,F5.1)
980   FORMAT (//48X,35HBREAK POINTS FOR THIS CONFIGURATION//)
990   FORMAT (28X,5HPOINT,6X,1HX,11X,1HY,11X,5HSWEEP,10X,8HDIHEDRAL,7X,4
     1HMOVE/38X,3HREF,9X,3HREF,10X,5HANGLE,11X,5HANGLE,9X,4HCODE/)
 1000 FORMAT(/52X,A10,22H PLANFORM BREAK POINTS/)
1010  FORMAT (////25X,34HTHE BREAKPOINT LOCATED SPANWISE AT,F11.5,3X,20H
     1HAS BEEN ADJUSTED TO,F9.5////)
 1020 FORMAT(1H0,43X,I5,
     $ 41H HORSESHOE VORTICES IN EACH CHORDWISE ROW )
1030  FORMAT (/23X,96HTABLE OF HORSESHOE VORTICES IN EACH CHORDWISE ROW
     1(FROM TIP TO ROOT BEGINNING WITH FIRST PLANFORM)//25F5.0/25F5.0)
C
 1040 FORMAT(///,53X,
     $ 30HHORSESHOE VORTEX SUMMARY TABLE  ,/,
     $ 33X, I5, 37H HORSESHOE VORTICIES USED ON THE LEFT,
     $ 27H HALF OF THIS CONFIGURATION ,//,
     $ 50X,8HPLANFORM,7X,5HTOTAL,8X,8HSPANWISE,/)
C
1050  FORMAT (52X,I4,10X,I3,11X,I4)
 1060 FORMAT(1H1, 9X, 22HF A T A L    E R R O R  ,//,
     $ 10X, 46HYOU ARE REQUESTING A VORTEX LIFT SOLUTION FOR  ,/,
     $ 10X, 46HA CAMBERED WING WITH -CLDES- NOT EQUAL TO 100. ,/,
```

```
      $ 10X, 46HTHIS WILL NOT YIELD ACCURATE RESULTS. PLEASE   ,/,         GEOMT680
      $ 10X, 46HRESUBMIT WITH  -CLDES-  = 100.                      )      GEOMT681
 1070 FORMAT(1H1,9X,22HF A T A L     E R R O R  ,/,                        GEOMT682
      $ 10X,39HTHE VALUE OF -VIC- IS TOO SCALL. PLEASE ,/,                 GEOMT683
      $ 10X,30HMAKE CORRECTIONS AND RESUBMIT. )                            GEOMT684
C                                                                          GEOMT685
 1080 FORMAT(1H1,9X,22HF A T A L     E R R O R  ,/,                        GEOMT686
      $ 10X,34HTHE VALUE OF -SCW- MUST BE GREATER ,/,                      GEOMT687
      $ 10X,28HTHAN 2 FOR THIS COMBINATION.  )                             GEOMT688
C                                                                          GEOMT689
 1090 FORMAT(1H1,9X,22HF A T A L     E R R O R ,/,                         GEOMT690
      $ 10X,*YOUR INPUT VALUES FOR -STA- DID NOT *,/,                      GEOMT691
      $ 10X,*MATCH UP WITH THE NUMBER OF SPANWISE*,/,                      GEOMT692
      $ 10X,*STATIONS COMPUTED. REFEP TO THE HORSESHOE *,/,                GEOMT693
      $ 10X,*VORTEX SUMMARY TABLE ABOVE TO OBTAIN THE *,/,                 GEOMT694
      $ 10X,*PROPER VALUES FOR -STA- PER PLANFORM*,/)                      GEOMT695
C                                                                          GEOMT696
 1100 FORMAT(1H1,9X,22HF A T A L     E R R O R ,/,                         GEOMT697
      $ 10X,*YOU HAVE REQUESTED THE FLOWFIELD OPTION, BUT*,/,              GEOMT698
      $ 10X,*HAVE SPECIFIED -CLDES- INCORRECTLY (*,F5.1,*) *,/,            GEOMT699
      $ 10X, *PLEASE MAKE CORRECTIONS AND RESUBMIT. *,/)                   GEOMT700
C                                                                          GEOMT701
C                                                                          GEOMT702
 1110 FORMAT(1H1,9X,22HF A T A L     E R R O R ,/,                         GEOMT703
      $ 10X,*THE SIDE-EDGE FORCE IS NOT PROPERLY COMPUTED *)               GEOMT704
2000  CONTINUE                                                             GEOMT705
      END                                                                  GEOMT706
      SUBROUTINE PLANPLT(IPLAN,XX,YY,AN,ICON,ITHETA)                       PLANPLT2
C                                                                          PLANPLT3
C                THIS ROUTINE PREPARES A PLOT DIAGRAM OF THE               PLANPLT4
C                INPUT PLANFORM CONFIGURATION(S) FOR THE LINE              PLANPLT5
C                PRINTER                                                   PLANPLT6
C                                                                          PLANPLT7
C              IPLAN--NUMBER OF PLANFORMS TO BE PLOTTED                    PLANPLT8
C      XX  --X COORDINATE ARRAY DIMENSIONED 25 X 4                         PLANPLT9
C      YY  --Y COORDINATE ARRAY DIMENSIONED 25 X 4                         PLANPL10
C              AN   --NUMBER OF POINTS IN PLANFORM LESS ONE                PLANPL11
C                                                                          PLANPL12
C              ICON --0 REQUESTS CONTOUR DRAWINGS.                         PLANPL13
C              ITHETA-ANGLE OF ROTATION DESIRED IN DEGREES,                PLANPL14
C                     0 FOR NO ROTATION                                    PLANPL15
C                                                                          PLANPL16
C                ROBERT GRAY  COMPUTER SCIENCES CORP  1980                 PLANPL17
C                                                                          PLANPL18
      DIMENSION XX(25,4), YY(25,4), AN(4)                                  PLANPL19
      DIMENSION NARRAY (100)                                               PLANPL20
      DIMENSION XXX(25,4), YYY(25,4)                                       PLANPL21
      DIMENSION NXX(26,4), NYY(26,4), NCHAR(4)                             PLANPL22
      DIMENSION IXPTS(50)                                                  PLANPL23
```

```
C                                                                       PLANPL24
      DATA NARRAY /100*1H /                                             PLANPL25
      DATA IPLUS /1H+/                                                  PLANPL26
      DATA IBLANK /10H          /                                       PLANPL27
      DATA NCHAR/1H@,1H#,1H$,1H&/                                       PLANPL28
      DATA YMIN /99999.0/, YMAX /-99999.0/                              PLANPL29
      DATA XMIN /99999.0/, XMAX /-99999.0/                              PLANPL30
C                                                                       PLANPL31
C                                                                       PLANPL32
C        ROTATE PLANFORM(S)                                             PLANPL33
C                                                                       PLANPL34
      THETA = ITHETA * (3.14159/180)                                    PLANPL35
      CANS = COS (THETA)                                                PLANPL36
      SANS = SIN (THETA)                                                PLANPL37
      DO 5 I = 1, IPLAN                                                 PLANPL38
        M = AN(I) + 1                                                   PLANPL39
        DO 5 N = 1,M                                                    PLANPL40
          XXX(N,I) = XX(N,I) * CANS - YY(N,I) * SANS                    PLANPL41
          YYY(N,I) = XX(N,I) * SANS + YY(N,I) * CANS                    PLANPL42
    5 CONTINUE                                                          PLANPL43
C                                                                       PLANPL44
C             RESCALE ALL X AND Y COORDINATES                           PLANPL45
C             Y IS 6 PER INCH ON THE LINE PRINTER, MAX 7 INCHES         PLANPL46
C             X IS 10 PER INCH, MAX 10 INCHES                           PLANPL47
C                                                                       PLANPL48
C             FIND MINIMUM AND MAXIMUM COORDINATES                      PLANPL49
      DO 10 I = 1,IPLAN                                                 PLANPL50
        M = AN(I) + 1                                                   PLANPL51
        DO 20 N = 1,M                                                   PLANPL52
          IF (YYY(N,I) .LT. YMIN) YMIN = YYY(N,I)                       PLANPL53
          IF (YYY(N,I) .GT. YMAX) YMAX = YYY(N,I)                       PLANPL54
          IF (XXX(N,I) .LT. XMIN) XMIN = XXX(N,I)                       PLANPL55
          IF (XXX(N,I) .GT. XMAX) XMAX = XXX(N,I)                       PLANPL56
   20   CONTINUE                                                        PLANPL57
   10 CONTINUE                                                          PLANPL58
C               VALUES USED AS SUBSCRIPTS SHOULD BE GREATER             PLANPL59
C               THAN OR EQUAL TO 1                                      PLANPL60
C                                                                       PLANPL61
      IF (YMIN .GT. 0.0) YUP= 0.0 - YMIN                                PLANPL62
      IF (YMIN .LE. 0.0) YUP = ABS(YMIN)                                PLANPL63
        DO 70 I = 1,IPLAN                                               PLANPL64
          M = AN(I) + 1                                                 PLANPL65
          DO 80 N = 1,M                                                 PLANPL66
          YYY(N,I) = YYY(N,I) + YUP + 1.0                               PLANPL67
   80     CONTINUE                                                      PLANPL68
   70 CONTINUE                                                          PLANPL69
      YMAX = YMAX + YUP + 1                                             PLANPL70
      YMIN = YMIN + YUP + 1.0                                           PLANPL71
C                                                                       PLANPL72
```

```
C                                                                       PLANPL73
      IF (XMIN .GT. 0.0) XUP = 0.0 - XMIN                               PLANPL74
      IF (XMIN .LE. 0.0) XUP = ABS(XMIN)                                PLANPL75
        DO 85 I = 1,IPLAN                                               PLANPL76
          M = AN(I) + 1                                                 PLANPL77
          DO 90 N = 1,M                                                 PLANPL78
          XXX(N,I) = XXX(N,I) + XUP + 1.0                               PLANPL79
 90     CONTINUE                                                        PLANPL80
 85    CONTINUE                                                         PLANPL81
      XMAX = XMAX + XUP + 1.0                                           PLANPL82
      XMIN = XMIN + XUP + 1.0                                           PLANPL83
C                                                                       PLANPL84
C             FIND Y SCALING FACTOR--MAXIMUM 42 LINES                   PLANPL85
      YDIM = YMAX                                                       PLANPL86
      YSCALE = 42.0/YDIM                                                PLANPL87
C             FIND X SCALING FACTOR--10 CHARACTERS PER INCH             PLANPL88
C             AND IN PROPORTION TO Y INCHES                             PLANPL89
      XSCALE = YSCALE * 1.666                                           PLANPL90
      XDIM = XMAX * XSCALE                                              PLANPL91
      IF (XDIM .LE. 100.49) GO TO 30                                    PLANPL92
C             SCALED X DIMENSION IS GREATER THAN 100, RESCALE           PLANPL93
C                                                                       PLANPL94
      RESCALE = 100.0/XDIM                                              PLANPL95
      YSCALE = YSCALE * RESCALE                                         PLANPL96
      XSCALE = XSCALE * RESCALE                                         PLANPL97
C                                                                       PLANPL98
C             COMPUTE ALL RESCALED X AND Y COORDINATES,                 PLANPL99
C             ROUNDING EACH                                             PLANP100
 30   DO 40 I = 1,IPLAN                                                 PLANP101
        M = AN(I) + 1                                                   PLANP102
        DO 50 N = 1,M                                                   PLANP103
          RXX = XXX(N,I) * XSCALE * 10.0 + 5.0                          PLANP104
          NXX(N,I) = RXX/10.0                                           PLANP105
          RYY = YYY(N,I) * YSCALE * 10.0 + 5.0                          PLANP106
          NYY(N,I) = RYY/10.0                                           PLANP107
 50     CONTINUE                                                        PLANP108
 40   CONTINUE                                                          PLANP109
C                                                                       PLANP110
C             FIND INTEGER MAXIMA AND MINIMA                            PLANP111
C                                                                       PLANP112
      YMAX  = YMAX *  YSCALE * 10.0 + 5.0                               PLANP113
      MAXYY = YMAX/10.0                                                 PLANP114
      YMIN  = YMIN * YSCALE * 10.0 + 5.0                                PLANP115
      MINYY = YMIN/10                                                   PLANP116
      XMAX = XMAX * XSCALE * 10.0 + 5.0                                 PLANP117
      MAXXX = XMAX/10.0                                                 PLANP118
      XMIN = XMIN * XSCALE * 10.0 + 5.0                                 PLANP119
      MINXX = XMIN/10.0                                                 PLANP120
C                                                                       PLANP121
```

```
C               CENTER THE DATA
C
      ICNTER = ((100 - (MAXXX - MINXX)) /2) - MINXX
        DO 65 I = 1,IPLAN
          M = AN(I) + 1
          DO 65 N = 1,M
            NXX(N,I) = NXX(N,I) + ICNTER
 65     CONTINUE
C
C
C             WRITE HEADING
  60  WRITE(6,802) (I,NCHAR(I),I=1,IPLAN)
  802 FORMAT(1H1,51X,34HAPPROXIMATE PLANFORM CONFIGURATION,/
     *(4X,9HPLANFORM ,I2,6H  IS  ,A1))
C
C               DRAW THE INPUT PLANFORM CONFIGURATIONS
C
C              FOR EACH SCAN LINE
C
      DO 500 ISCAN = MINYY,MAXYY
C              CARRIAGE CONTROL FOR THE FIRST PLANFORM ON A LINE
C              IS BLANK, AFTERWARDS IT IS A PLUS
      ICC = IBLANK
C
C             FOR EACH PLANFORM
C             FIND POINTS ON THIS SCAN LINE
C
      DO 400 I = 1,IPLAN
        M = AN(I) + 1
C              FIND BREAKPOINTS ON THIS LINE
        DO 110 N = 1,M
          IF (NYY(N,I) .EQ. ISCAN) NARRAY (NXX(N,I)) = NCHAR(I)
 110    CONTINUE
C
C              INSURE A CLOSED PLANFORM
      NXX(M+1,I) = NXX(1,I)
      NYY(M+1,I) = NYY(1,I)
C
C             OFFSET SCAN LINE TO INSURE VERTICES ARE NOT INTERSECTED
C
      SCAN = ISCAN + 0.1
C
C             FIND POINTS AT WHICH SCAN LINE INTERSECTS LINE
C             SEGMENTS
C
      ICOUNT = 0
C             FOR EACH LINE SEGMENT...
      DO 300 N = 1,M
C             IF THIS SEGMENT IS HORIZONTAL, IT MUST BE
```

```
C             FILLED SEPARATELY.                                        PLANP171
C                                                                       PLANP172
      IF (ISCAN .NE. NYY(N,I)) GO TO 120                                PLANP173
      IF (NYY(N+1,I) .NE. NYY(N,I)) GO TO 120                           PLANP174
        ILEFT = MINO(NXX(N,I),NXX(N+1,I))                               PLANP175
        IRIGHT = MAXO(NXX(N,I),NXX(N+1,I))                              PLANP176
        DO 130 J = ILEFT,IRIGHT                                         PLANP177
          NAPRAY(J) = NCHAR(I)                                          PLANP178
 130    CONTINUE                                                        PLANP179
        ICOUNT = ICOUNT + 1                                             PLANP180
        MAXNXX = MAXO(NXX(N,I),NXX(N+1,I))                              PLANP181
        MINNXX = MINO(NXX(N,I),NXX(N+1,I))                              PLANP182
        IXPTS(ICOUNT) = MINNXX                                          PLANP183
        ICOUNT = ICOUNT + 1                                             PLANP184
        IXPTS(ICOUNT) = MAXNXX                                          PLANP185
C                                                                       PLANP186
 120    RYMAX = AMAXO(NYY(N,I),NYY((N+1),I))                            PLANP187
        RYMIN = AMINO(NYY(N,I),NYY((N+1),I))                            PLANP188
        IF (SCAN .LT. RYMIN .OR. SCAN .GT. RYMAX ) GO TO 300            PLANP189
C                                                                       PLANP190
C            THE SCAN LINE WILL INTERSECT                               PLANP191
C            THIS SEGMENT                                               PLANP192
C                                                                       PLANP193
        A = SCAN - NYY((N+1),I)                                         PLANP194
        B = NXX((N+1),I) - NXX(N,I)                                     PLANP195
        C = NYY((N+1),I) - NYY(N,I)                                     PLANP196
C            IF SEGMENT IS HORIZONTAL,                                  PLANP197
C            IGNORE IT                                                  PLANP198
        IF (NYY((N+1),I) .EQ. NYY(N,I)) GO TO 300                       PLANP199
        D =A * B/C                                                      PLANP200
C                                                                       PLANP201
C            THE POINT OF INTERSECTION IS                               PLANP202
C            (IPTINT,SCAN)                                              PLANP203
C                                                                       PLANP204
        RPTINT = (D + NXX(N + 1,I)) * 10.0 + 5.0                        PLANP205
        IPTINT = RPTINT/10.0                                            PLANP206
        MAXNXX = MAXO(NXX(N,I),NXX((N+1),I))                            PLANP207
        MINNXX = MINO(NXX(N,I),NXX((N+1),I))                            PLANP208
C                                                                       PLANP209
C            IF INTERSECTION IS TO THE LEFT OR                          PLANP210
C            RIGHT OF ENDPOINTS, IGNORE IT                              PLANP211
C                                                                       PLANP212
        IF (IPTINT .GT. MAXNXX .OR. IPTINT .LT. MINNXX) GO TO 300       PLANP213
C                                                                       PLANP214
C            STORE AND COUNT IPTINT                                     PLANP215
C                                                                       PLANP216
        ICOUNT = ICOUNT + 1                                             PLANP217
        IXPTS(ICOUNT) = IPTINT                                          PLANP218
 300  CONTINUE                                                          PLANP219
```

```
C
C             IF ONLY ONE INTERSECTION
C             FOUND, GO TO NEXT PLANFORM OR SCANLINE
C
      IF (ICOUNT .LE. 1) GO TO 400
C
C             SORT THE INTERSECTIONS. THEY
C             WILL BE FILLED LEFT TO RIGHT.
C
      ICT = ICOUNT - 1
      DO 310 K = 1,ICT
        INDX = ICT - K + 1
        DO 310 J = 1,INDX
          IF (IXPTS(J) .LT. IXPTS(J+1)) GO TO 310
          ITMP = IXPTS(J)
          IXPTS(J) = IXPTS(J+1)
          IXPTS(J+1) = ITMP
 310  CONTINUE
C      CONTOUR DRAWING REQUESTED, PLACE
C          INTERSECTION POINTS IN NARRAY
C
      IF (ICON .NE. 0) GO TO 320
      DO 315 N = 1,ICOUNT
        NARRAY(IXPTS(N)) = NCHAR(I)
 315  CONTINUE
      GO TO 340
C
C      CONTRAST DRAWING REQUESTED, FILL IN
C          BETWEEN INTERSECTION POINTS
C
 320  DO 330 N= 1,ICOUNT,2
        ILEFT = IXPTS(N)
        IRIGHT = IXPTS (N+1)
        DO 330 IX = ILEFT,IRIGHT
          NARRAY(IX) = NCHAR(I)
 330  CONTINUE
C
C          WRITE THE LINE
 340  WRITE(6,804) ICC,NARRAY
 804  FORMAT (A1,18X,100A1)
C          PREPARE TO OVERWRITE THE NEXT PLANFORM
      ICC = IPLUS
C          CLEAR NARRAY FOR NEXT PLANFORM
      DO 350 N = 1,100
        NARRAY(N) = IBLANK
 350  CONTINUE
C
 400  CONTINUE
C
```

```
 500  CONTINUE                                                       PLANP269
C                                                                    PLANP270
      RETURN                                                         PLANP271
      END                                                            PLANP272
      SUBROUTINE MATXSOL                                             MATXSOL2
      DIMENSION YY(2),FV(2),FW(2),FVN(400)                           MATXSOL3
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,             ALL     2
     $    STA(4), TBLSCW(100), YYCP(4),                              ALL     3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),      ALL     4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                     ALL     5
C                                                                    ALL     6
      COMMON /TOTHREE/ CIR(400,2)                                    TOTHRE  2
C                                                                    TOTHRE  3
      COMMON /INSUB23/ APSI,APHI,XX,YYY,ZZ,SNN,TOLC                  MATXSOL6
      DIMENSION WA(20000), INDEX(2)                                  MATXSOL7
C                                                                    MATXSOL8
C                                                                    MATXSOL9
C     PART 2 - COMPUTE CIRCULATION TERMS                             MATXSO10
C                                                                    MATXSO11
C                                                                    MATXSO12
C     THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE  MATXSO13
C     CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES           MATXSO14
C                                                                    MATXSO15
C                                                                    MATXSO16
      TOLC=(BOT*15.E-05)**2                                          MATXSO17
      DO 10 NV=1,M                                                   MATXSO18
      CIR(NV,1)=12.5663704*ALP(NV)                                   MATXSO19
      CIR(NV,2)=12.5663704                                           MATXSO20
      IF (PTEST.NE.0.) CIR(NV,2)=-1.0964155*Q(NV)/BOT                MATXSO21
      IF (QTEST.NE.0.) CIR(NV,2)=-1.0964155*PV(NV)*BETA              MATXSO22
 10   CONTINUE                                                       MATXSO23
      IZZ=1                                                          MATXSO24
      NNV=TBLSCW(IZZ)                                                MATXSO25
      REWIND 10                                                      MATXSO26
      DO 70 NV=1,M                                                   MATXSO27
      DO 20 I=1,M                                                    MATXSO28
 20   FVN(I)=0.                                                      MATXSO29
      IZ=1                                                           MATXSO30
      NNN=TBLSCW(IZ)                                                 MATXSO31
      DO 60 NN=1,M                                                   MATXSO32
      APHI=ATAN(PHI(IZ))                                             MATXSO33
      APSI=PSI(NN)                                                   MATXSO34
      XX=PV(NV)-PN(NN)                                               MATXSO35
      YY(1)=Q(NV)-Q(NN)                                              MATXSO36
      YY(2)=Q(NV)+Q(NN)                                              MATXSO37
      ZZ=ZH(IZZ)-ZH(IZ)                                              MATXSO38
      SNN=S(NN)                                                      MATXSO39
      DO 30 I=1,2                                                    MATXSO40
      YYY=YY(I)                                                      MATXSO41
```

```
      CALL INFSUB (BOT,FV(I),FW(I),FUI)                            MATXS042
      APHI=-APHI                                                   MATXS043
      APSI=-APSI                                                   MATXS044
30    CONTINUE                                                     MATXS045
      IF (PTEST.NE.0.) GO TO 40                                    MATXS046
      FVN(NN)=FW(1)+FW(2)-(FV(1)+FV(2))*PHI(IZZ)                   MATXS047
      GO TO 50                                                     MATXS048
40    FVN(NN)=FW(1)-FW(2)-(FV(1)-FV(2))*PHI(IZZ)                   MATXS049
50    IF (NN.LT.NNN.OR.NN.EQ.M) GO TO 60                           MATXS050
      IZ=IZ+1                                                      MATXS051
      NNN=NNN+TBLSCW(IZ)                                           MATXS052
60    CONTINUE                                                     MATXS053
      DUMB=-CIR(NV,1)                                              MATXS054
      DUMY=-CIR(NV,2)                                              MATXS055
      WRITE(10) (FVN(I),I=1,M)                                     MATXS056
      IF (NV.LT.NNV.OR.NV.EQ.M) GO TO 70                           MATXS057
      IZZ=IZZ+1                                                    MATXS058
      NNV=NNV+TBLSCW(IZZ)                                          MATXS059
70    CONTINUE                                                     MATXS060
      LWA = 20000                                                  MATXS061
      CALL GIVENS(M,M,2,WA,LWA,CIR,10,11,IERR)                     MATXS062
      CALL STINDX(11,INDEX,1,0)                                    MATXS063
      END                                                          MATXS064
      SUBROUTINE GIVENS(NR,NC,NS,W,KORE,B,LSEQ,LRAF,FLAG)          GIVENS 2
C                                                                  GIVENS 3
C SOLVE A LARGE SET OF LINEAR EQUATIONS OF THE FORM AX = B         GIVENS 4
C IN A SMALL CENTRAL MEMORY WORKING AREA USING GIVENS              GIVENS 5
C TRIANGULARIZATION.  THE SET MAY BE OVERDETERMINED.  THE          GIVENS 6
C A-MATRIX IS STORED BY ROWS ON A SEQUENTIAL FILE. THE WORK-       GIVENS 7
C ING AREA IS DYNAMICALLY MANAGED USING A RANDOM ACCESS            GIVENS 8
C FILE CREATED WITHIN THE MODULE.                                  GIVENS 9
C                                                                  GIVENS10
C ON ENTRY                                                         GIVENS11
C   NR   = NUMBER OF ROWS OF A.                                    GIVENS12
C   NC   = NUMBER OF COLUMNS OF A.                                 GIVENS13
C   NS   = NUMBER OF RIGHT HAND SIDES.                             GIVENS14
C   W    = WORKING AREA ARRAY.                                     GIVENS15
C   KORE = LENGTH OF THE WORKING AREA, GE 4 * (NC + NS)            GIVENS16
C          <IF NEGATIVE, KORE IS SET TO [FL - W(1)] >              GIVENS17
C   B    = NR * NS MATRIX OF RIGHT HAND SIDES.                     GIVENS18
C   LSEQ = LOGICAL UNIT NUMBER OF FILE HOLDING ROWS OF A.          GIVENS19
C   LRAF = LOGICAL UNIT NUMBER OF RANDOM ACCESS FILE.              GIVENS20
C          (GIVENS SUBINDEXES THIS FILE.  THE USER                 GIVENS21
C          MUST RESTORE THE MASTER INDEX UPON RETURN)              GIVENS22
C                                                                  GIVENS23
C UPON RETURN                                                      GIVENS24
C   B HOLDS THE NS SOLUTION VECTORS, EACH STORED OVER THE          GIVENS25
C          FIRST NC WORDS OF THE CORRESPONDING COLUMN.             GIVENS26
C   FLAG = 0, EXECUTION PROCEEDED NORMALLY.                        GIVENS27
```

```
C          = -, WORKING AREA TOO SMALL, -(FLAG) WORDS NEEDED.
C          = +, MATRIX SINGULAR, FLAG = NUMBER OF SINGULAR ROW
C
C C.W.BOLZ AND R.W.HAMM, COMPUTER SCIENCES CORP., 1974.
C
      DIMENSION W(KORE),B(400,2)
      COMMON/HDATAR/ KM,NBLK,NRAF,NROWS,NSEQ,TOL
      INTEGER FLAG
C INITIALIZE
      DO 4 I=1,KORE
    4   W(I)=0.0
      FLAG=0
      NSEQ=LSEQ
      NRAF=LRAF
      NCS=NC+NS
      TOL=1.0E-100
C SIZE THE WORKING AREA.
      NWA=4*(NC+NS)
      IF(KORE.GE.NWA) GO TO 1
      FLAG=-NWA
      RETURN
C
C PARTITION THE WORKING AREA. NROWS IS NUMBER OF ROWS TO BE
C PROCESSED PER REDUCTION PASS. KM IS PORTION OF WORKING
C AREA AVAILABLE FOR HOUSEKEEPING ARRAYS AND SCRATCH STORAGE
C NBLK IS NUMBER OF BLOCKS OF THE TRIANGULARIZED MATRIX.
    1 NROWS=KORE/(2*NCS)+1
    2 KM=KORE-NROWS*NCS
      NBLK=NC*NCS/(2*KM)+1
      KM=KM-3*NBLK-2
      IF(KM.GE.2*NCS-1) GO TO 3
C NROWS OVERESTIMATED. REDUCE.
      NROWS=NROWS-1
      GO TO 2
C
C COMPUTE PARTITION (BLOCK) SIZES AND NUMBER OF ROWS IN EACH
    3 NB=NBLK
      CALL BLOCKR(W,NB,NC,NCS,KORE)
C
C NOW DYNAMICALLY ALLOCATE THE WORKING AREA...
      LB=1
      LI=LB+2*NBLK
      LR=LI+NBLK+2
      LW=LR+NROWS*NCS
C
C SUBINDEX THE RANDOM ACCESS FILE.
      CALL STINDX(NRAF,W(LI),NBLK+1,0)
C
C AND SOLVE USING GIVENS TRIANGULARIZATION.
```

```
      CALL TRIANG(W(LB),W(LR),W(LW),B,NR,NC,NS,NCS,NBLK,             GIVENS77
     .            NROWS,KM,FLAG)                                      GIVENS78
C                                                                     GIVENS79
      RETURN                                                          GIVENS80
      END                                                             GIVENS81
      SUBROUTINE BLOCKR(LBLK,NB,NC,NCS,KORE)                          GIVENS82
C                                                                     GIVENS83
C COMPUTE SIZE OF, NUMBER OF ROWS IN, EACH PARTITION                  GIVENS84
C OF THE TRIANGULARIZED MATRIX.                                       GIVENS85
C                                                                     GIVENS86
C R.W.HAMM, COMPUTER SCIENCES CORP., 1974                             GIVENS87
C                                                                     GIVENS88
      DIMENSION LBLK(2,NB)                                            GIVENS89
      COMMON/HDATAR/ KM,NBLK,NRAF,NROWS,NSEQ,TOL                      GIVENS90
    1 NT=NCS                                                          GIVENS91
      IC=NC                                                           GIVENS92
      NBLK=0                                                          GIVENS93
      MAXW=0                                                          GIVENS94
C                                                                     GIVENS95
C LOOP ON BLOCKS OF MATRIX.                                           GIVENS96
    2 NBLK=NBLK+1                                                     GIVENS97
      KW=KM                                                           GIVENS98
      LNG=0                                                           GIVENS99
      IRW=0                                                           GIVEN100
C                                                                     GIVEN101
C COMPUTE ROWS WITHIN BLOCK. IRW = NO.OF ROWS, LNG = LENGTH           GIVEN102
C OF BLOCK, NT = LENGTH OF ROW I, KW = WORDS REMAINING.               GIVEN103
      DO 3 I=1,IC                                                     GIVEN104
        KW=KW-NT                                                      GIVEN105
        IF(KW.LT.0) GO TO 4                                           GIVEN106
        LNG=LNG+NT                                                    GIVEN107
        IRW=IRW+1                                                     GIVEN108
        NT=NT-1                                                       GIVEN109
    3   CONTINUE                                                      GIVEN110
    4 LBLK(1,NBLK)=LNG                                                GIVEN111
      LBLK(2,NBLK)=IRW                                                GIVEN112
      MAXW=MAX0(LNG,MAXW)                                             GIVEN113
C                                                                     GIVEN114
C NEXT BLOCK                                                          GIVEN115
      IC=IC-IRW                                                       GIVEN116
      IF(IC.GT.0) GO TO 2                                             GIVEN117
C                                                                     GIVEN118
C TEST IF WORKING AREA OVERRUN. MAXW = MAX.PARTITION SIZE.            GIVEN119
      NWORDS=MAXW+NCS*NROWS+3*NBLK+2                                  GIVEN120
      NDIF=KORE-NWORDS                                                GIVEN121
      IF(NDIF) 5,7,6                                                  GIVEN122
C WORKING AREA OVERRUN. REDUCE BLOCK SIZE AND REITERATE.              GIVEN123
    5 KM=KM+NDIF                                                      GIVEN124
      GO TO 1                                                         GIVEN125
```

```
C UNDERESTIMATED. CAN WE INCREASE NROWS...                               GIVEN126
    6 NROWS=NROWS+NDIF/NCS                                               GIVEN127
      KM=MAXW                                                            GIVEN128
    7 RETURN                                                             GIVEN129
      END                                                                GIVEN130
      SUBROUTINE TRIANG(LBLK,ROWS,W,B,NR,NC,NS,NCS,NB,                   TRIANG 2
     .                  NRW,KMAX,FLAG)                                   TRIANG 3
C                                                                        TRIANG 4
C TRIANGULARIZE THE AUGMENTED MATRIX USING PLANE                         TRIANG 5
C ROTATIONS, THEN BACK-SUBSTITUTE FOR SOLUTIONS.                         TRIANG 6
C                                                                        TRIANG 7
C R.W.HAMM AND C.W.BOLZ, COMPUTER SCIENCES CORP., 1974.                  TRIANG 8
C                                                                        TRIANG 9
      DIMENSION LBLK(2,NB),ROWS(NCS,NRW),W(KMAX),B(400,2)                TRIANG10
      COMMON/HDATAR/ KM,NBLK,NRAF,NROWS,NSEQ,TOL                         TRIANG11
      INTEGER FLAG                                                       TRIANG12
C                                                                        TRIANG13
C INITIALIZE SCRATCH FILE NRAF TO SIMPLIFY LATER LOGIC.                  TRIANG14
      DO 1 I=1,KM                                                        TRIANG15
    1   W(I)=0.0                                                         TRIANG16
      DO 2 N=1,NBLK                                                      TRIANG17
        LNG=LBLK(1,N)                                                    TRIANG18
        CALL WRITMS(NRAF,W,LNG,N)                                        TRIANG19
    2   CONTINUE                                                         TRIANG20
      NCSP=NCS+1                                                         TRIANG21
      IR=0                                                               TRIANG22
C                                                                        TRIANG23
C TRIANGULARIZATION LOOP. READ NROWS ROWS PER PASS, AUGMENT.             TRIANG24
      REWIND NSEQ                                                        TRIANG25
      DO 15 IRSET=1,NR,NROWS                                             TRIANG26
        DO 4 JRR=1,NROWS                                                 TRIANG27
          CALL BUFFIN(NSEQ,ROWS(1,JRR),NC,IEOF)                          TRIANG28
          IF(IEOF.EQ.0) GO TO 5                                          TRIANG29
          JR=JRR                                                         TRIANG30
          IS=IR+JR                                                       TRIANG31
          DO 3 J=1,NS                                                    TRIANG32
            NCJ=NC+J                                                     TRIANG33
    3       ROWS(NCJ,JR)=B(IS,J)                                         TRIANG34
    4     CONTINUE                                                       TRIANG35
    5   IRR=IR+JR                                                        TRIANG36
        IR=IR+1                                                          TRIANG37
        NRWZ=0                                                           TRIANG38
C                                                                        TRIANG39
C READ IN A BLOCK AND SET ITS ROW INDICES.                               TRIANG40
        DO 13 N=1,NBLK                                                   TRIANG41
          LNG=LBLK(1,N)                                                  TRIANG42
          NRWA=NRWZ+1                                                    TRIANG43
          NRWZ=LBLK(2,N)+NRWZ                                            TRIANG44
          IF(IRR.LT.NRWA) GO TO 14                                       TRIANG45
```

```
          CALL READMS(NRAF,W,LNG,N)                                  TRIANG46
C                                                                     TRIANG47
C LOOP THROUGH ROWS READ IN THIS PASS,                               TRIANG48
          LR=0                                                       TRIANG49
          DO 12 I=IR,IRR                                             TRIANG50
            LR=LR+1                                                  TRIANG51
            JJ=1                                                     TRIANG52
C AND PROCESS ROWS OF THIS BLOCK.                                    TRIANG53
            DO 11 J=NRWA,NRWZ                                        TRIANG54
              IF(I-J) 12,9,6                                         TRIANG55
    6         AB=ROWS(J,LR)                                          TRIANG56
              IF(ABS(AB).LT.TOL) GO TO 8                             TRIANG57
              R=SQRT(AB*AB+W(JJ)*W(JJ))                              TRIANG58
              IF(R.LT.TOL) GO TO 8                                   TRIANG59
              C=W(JJ)/R                                              TRIANG60
              S=AB/R                                                 TRIANG61
              W(JJ)=R                                                TRIANG62
              JP=J+1                                                 TRIANG63
              KK=JJ+1                                                TRIANG64
C                                                                     TRIANG65
C ELIMINATE ELEMENT J OF INPUT ROW.                                  TRIANG66
              DO 7 K = JP,NCS                                        TRIANG67
                T = C*W(KK) + S*ROWS(K,LR)                           TRIANG68
                ROWS(K,LR) = C*ROWS(K,LR) - S*W(KK)                  TRIANG69
                W(KK) = T                                            TRIANG70
                KK = KK + 1                                          TRIANG71
    7         CONTINUE                                               TRIANG72
C                                                                     TRIANG73
    8         CONTINUE                                               TRIANG74
              JJ=JJ+NCSP-J                                           TRIANG75
              GO TO 11                                               TRIANG76
C ELEMENTS J TO NCS OF ROWS TO WORKING AREA.                         TRIANG77
    9         KK=JJ                                                  TRIANG78
              P=SIGN(1.0,ROWS(J,LR))                                 TRIANG79
              DO 10 K=J,NCS                                          TRIANG80
                W(KK)=R*ROWS(K,LR)                                   TRIANG81
   10           KK=KK+1                                              TRIANG82
   11         CONTINUE                                               TRIANG83
   12       CONTINUE                                                 TRIANG84
          CALL WRITMS(NRAF,W,LNG,N,1)                                TRIANG85
   13     CONTINUE                                                   TRIANG86
   14   IR=IRR                                                       TRIANG87
   15   CONTINUE                                                     TRIANG88
C                                                                     TRIANG89
C BACK-SUBSTITUTE FOR SOLUTION                                       TRIANG90
      CALL SOLVER(LBLK,W,B,NR,NC,NS,NB,KMAX,FLAG)                    TRIANG91
      RETURN                                                         TRIANG92
      END                                                            TRIANG93
      SUBROUTINE SOLVER(LBLK,W,B,NR,NC,NS,NB,KMAX,FLAG)              SOLVER 2
```

```
C                                                                       SOLVER 3
C BACK SUBSTITUTION FOR X. STORE OVER B.                                 SOLVER 4
C                                                                       SOLVER 5
C C.W.BOLZ AND R.W.HAMM, COMPUTER SCIENCES CORP., 1974.                  SOLVER 6
C                                                                       SOLVER 7
      DIMENSION LBLK(2,NB),W(KMAX),B(400,2)                             SOLVER 8
      COMMON/HDATAR/ KM,NBLK,NRAF,NROWS,NSEQ,TOL                        SOLVER 9
      INTEGER FLAG                                                      SOLVER10
C                                                                       SOLVER11
      LE=0                                                              SOLVER12
      NK=NBLK+1                                                         SOLVER13
      NSP=NS+1                                                          SOLVER14
      MR=NC                                                             SOLVER15
C                                                                       SOLVER16
C BLOCK LOOP. LAST BLOCK FIRST.                                         SOLVER17
      DO 6 NN=1,NBLK                                                    SOLVER18
        N=NK-NN                                                         SOLVER19
        LNG=LBLK(1,N)                                                   SOLVER20
        CALL READMS(NRAF,W,LNG,N)                                       SOLVER21
        NRW=LBLK(2,N)                                                   SOLVER22
        KK=LNG-NS                                                       SOLVER23
C                                                                       SOLVER24
C ROWS IN THIS BLOCK.                                                   SOLVER25
        DO 5 I=1,NRW                                                    SOLVER26
          DO 1 JS=1,NS                                                  SOLVER27
            KKS=KK+JS                                                   SOLVER28
    1       B(MR,JS)=W(KKS)                                             SOLVER29
C                                                                       SOLVER30
C SOLUTIONS                                                             SOLVER31
          DO 4 JS=1,NS                                                  SOLVER32
            TERM=0.0                                                    SOLVER33
            LN=LE                                                       SOLVER34
            LR=NC                                                       SOLVER35
            JJ=KK                                                       SOLVER36
C SUBSTITUTION                                                          SOLVER37
    2       IF(LN.EQ.0) GO TO 3                                         SOLVER38
              TERM=TERM+W(JJ)*B(LR,JS)                                  SOLVER39
              LR=LR-1                                                   SOLVER40
              LN=LN-1                                                   SOLVER41
              JJ=JJ-1                                                   SOLVER42
              GO TO 2                                                   SOLVER43
C TEST FOR SINGULAR ROW                                                 SOLVER44
    3       IF(W(JJ).LT.TOL) GO TO 7                                    SOLVER45
    4       B(LR,JS)=(B(LR,JS)-TERM)/W(JJ)                              SOLVER46
          MR=MR-1                                                       SOLVER47
          LE=LE+1                                                       SOLVER48
          KK=JJ-NSP                                                     SOLVER49
    5     CONTINUE                                                      SOLVER50
    6   CONTINUE                                                        SOLVER51
```

```
      RETURN                                                            SOLVER52
C                                                                       SOLVER53
C ERROR                                                                 SOLVER54
    7 FLAG=LR                                                           SOLVER55
      RETURN                                                            SOLVER56
      END                                                               SOLVER57
      SUBROUTINE BUFFIN(LUN,A,N,IER)                                    BUFFIN 2
      DIMENSION A(1)                                                    BUFFIN 3
      IER=N                                                             BUFFIN 4
      READ(LUN) (A(I),I=1,N)                                            BUFFIN 5
      IF(EOF(LUN)) 1,2                                                  BUFFIN 6
    1 IER=0                                                             BUFFIN 7
    2 RETURN                                                            BUFFIN 8
      END                                                               BUFFIN 9
```

```
      SUBROUTINE AERODYN                                                AERODYN2
C                                                                       AERODYN3
C                                                                       AERODYN4
      DIMENSION CLCC(400,2), CH(2,100), SUM(3), YCP(4), CROLL(4),       AERODYN5
     $ AC(4),CLCL(2,100),P(400),SMOAD(2,100),                           AERODYN6
     $SLDT(100),SMLD(2,100),SCT(5),SAT(5)                               AERODYN7
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                ALL    2
     $    STA(4), TBLSCW(100), YYCP(4),                                 ALL    3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),         ALL    4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                        ALL    5
C                                                                       ALL    6
      COMMON /TOTHREE/ CIR(400,2)                                       TOTHRE 2
C                                                                       TOTHRE 3
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD               THREFOR2
C                                                                       THREFOR3
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,    ONETHRE2
     $   ARTRUE, RTCOHT(4), CONFIG(2), NSSWSV(4),                       ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                               ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),      ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCIN,                          ONETHRE6
     $  YINNER(4), YOUTER(4)                                            ONETHRE7
C                                                                       ONETHRE8
      INTEGER CONFIG                                                    ONETHRE9
C                                                                       ONETHR10
      COMMON /THRECDI/ SLOAD(3,100)                                     AERODY12
      COMMON /INSUB23/ APSI,APHI,XXX,YYY,ZZZ,SNN,TOLC                   AERODY13
      COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4),  MAINONE2
     $  XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),       MAINONE3
     $  XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),    MAINONE4
     $  ZZ(25,4), ITIPCOD, ICAMTST                                      MAINONE5
C                                                                       MAINONE6
      DIMENSION NUMBER(4)                                               AERODY15
      DATA NUMBER/5HFIRST,6HSECOND,5HTHIRD,6HFOURTH/                    AERODY16
C                                                                       AERODY17
C                                                                       AERODY18
C     PART 3 - COMPUTE OUTPUT TERMS                                     AERODY19
C                                                                       AERODY20
C                                                                       AERODY21
      CLSAVE=CLDES                                                      AERODY22
      IF(CLDES.EQ.100.) CLDES=1.                                        AERODY23
```

```
      RAD=57.29578                                                      AERODY24
      ALREF=1                                                           AERODY25
      QINF=1.                                                           AERODY26
      TWST = 0.0                                                        AERODY27
      NSSW = 0                                                          AERODY28
      DO 5 IT = 1,IPLAN                                                 AERODY29
      TWST = TWST + TWIST(IT)                                           AERODY30
      NSSW = NSSW + NSSWSV(IT)                                          AERODY31
      CLWNG(IT) = 0.0                                                   AERODY32
      CLAMAR(IT) = 0.0                                                  AERODY33
      CLWIN(IT) = 0.0                                                   AERODY34
        YCP(IT) = 0.0                                                   AERODY35
        CROLL(IT) = 0.0                                                 AERODY36
        YYCP(IT) = 0.0                                                  AERODY37
    5 CONTINUE                                                          AERODY38
C                                                                       AERODY39
C                        PART 3 - SECTION 1                             AERODY40
C                COMPUTE LIFT AND PITCHING MOMENT HERE                  AERODY41
C                                                                       AERODY42
      IZ=1                                                              AERODY43
      NNN=TBLSCW(IZ)                                                    AERODY44
      DO 10 I=1,M                                                       AERODY45
      P(I)=S(I)*COS(ATAN(PHI(IZ)))                                      AERODY46
      IF (I.LT.NNN.OR.I.EQ.M) GO TO 10                                  AERODY47
      IZ=IZ+1                                                           AERODY48
      NNN=NNN+TBLSCW(IZ)                                                AERODY49
   10 CONTINUE                                                          AERODY50
      IT = 1                                                            AERODY51
      SUM(1) = 0.0                                                      AERODY52
      SUM(2) = 0.0                                                      AERODY53
      SUM(3) = 0.0                                                      AERODY54
      MSUM = MSV(IT)                                                    AERODY55
      DO 20 I=1,M                                                       AERODY56
      SUM(1) = SUM(1) + CIR(I,1) * P(I)                                 AERODY57
      SUM(2) = SUM(2) + CIR(I,2) * P(I)                                 AERODY58
      SUM(3) = SUM(3) + CIR(I,2) * P(I) * Q(I)                          AERODY59
      IF(I.EQ.MSUM) GOTO 15                                             AERODY60
      GOTO 20                                                           AERODY61
   15 CLWNG(IT) = SUM(1) * 8. / SREF                                    AERODY62
      CLWIN(IT) = SUM(2) * 8. / SREF                                    AERODY63
      CROLL(IT) = SUM(3) * 8. / (SREF*BOT)                              AERODY64
      IT = IT + 1                                                       AERODY65
      MSUM = MSUM + MSV(IT)                                             AERODY66
   20 CONTINUE                                                          AERODY67
      CLT=8.*SUM(1)/SREF                                                AERODY68
      CLNT=8.*SUM(2)/SREF                                               AERODY69
      IF (KBOT.EQ.1) GO TO 30                                           AERODY70
      CLWNGT = CLWNG(KBOT) - CLWNG(KBOT-1)                              AERODY71
      CLWING = CLWIN(KBOT) - CLWIN(KBOT-1)                              AERODY72
```

```
      GOTO 35                                                           AERODY73
   30 CLWNGT = CLWNG(1)                                                 AERODY74
      CLWING = CLWIN(1)                                                 AERODY75
   35 CRL = 0.0                                                         AERODY76
      DO 40 I=1,M                                                       AERODY77
      CRL=CRL+(Q(I)*CIR(I,2)*2.*P(I))*2.                                AERODY78
      CLCC(I,1)=CIR(I,1)*2/CAVE                                         AERODY79
   40 CLCC(I,2)=CIR(I,2)*2/CAVE                                         AERODY80
C                                                                       AERODY81
C     COMPUTE CLP                                                       AERODY82
C                                                                       AERODY83
      CLP=CRL/(SREF*BOT*0.08725)                                        AERODY84
      CLA(2)=CLNT                                                       AERODY85
      DO 120 IXX=1,2                                                    AERODY86
      SA=SB=SC=0.                                                       AERODY87
      I=0                                                               AERODY88
      IT = 0                                                            AERODY89
      JB=NSSWSV(1)                                                      AERODY90
      JA=1                                                              AERODY91
   50 CONTINUE                                                          AERODY92
      DO 70 JSSW=JA,JB                                                  AERODY93
      SD=SE=0.                                                          AERODY94
      SLOAD(IXX,JSSW)=0                                                 AERODY95
      NSCW=TBLSCW(JSSW)                                                 AERODY96
      DO 70 JSCW=1,NSCW                                                 AERODY97
      IF (TWST.EQ.0..AND.IXX.EQ.1) GO TO 60                             AERODY98
      I=I+1                                                             AERODY99
      SA=SA+CIR(I,IXX)*P(I)                                             AEROD100
      SB=SB+CIR(I,IXX)*Q(I)*P(I)                                        AEROD101
      SC=SC+CIR(I,IXX)*PN(I)*P(I)*BETA                                  AEROD102
      SLOAD(IXX,JSSW)=SLOAD(IXX,JSSW)+(BOT*CIR(I,IXX)*1.)/(2.*SUM       AEROD103
     1(IXX))                                                            AEROD104
      AA = P(I) / S(I)                                                  AEROD105
      SD = SD + CIR(I,IXX) * AA                                         AEROD106
      SE = SE + CIR(I,IXX) * PN(I) * BETA * AA                          AEROD107
      IF (JSCW.NE.NSCW) GO TO 70                                        AEROD108
      SMOAD(IXX,JSSW)=SE                                                AEROD109
      SMLD(IXX,JSSW)=SD                                                 AEROD110
      GO TO 70                                                          AEROD111
   60 SLOAD(1,JSSW)=SMOAD(1,JSSW)=SMLD(1,JSSW)=0.                       AEROD112
   70 CONTINUE                                                          AEROD113
      IT = IT + 1                                                       AEROD114
      IF (JSSW.GE.NSSW) GO TO 80                                        AEROD115
      JA = JA + NSSWSV(IT)                                              AEROD116
      JB = JB + NSSWSV(IT + 1)                                          AEROD117
      IF (IXX.EQ.1) GO TO 50                                            AEROD118
      IF(IT.EQ.1) GOTO 75                                               AEROD119
      SCT(IT) = SC                                                      AEROD120
      SAT(IT) = SA                                                      AEROD121
```

```
      CLAMAR(IT) = (SCT(IT) - SCT(IT-1)) / ((SAT(IT) - SAT(IT-1)) *CREF)AEROD122
      GOTO 50                                                           AEROD123
   75 SCT(IT) = SC                                                      AEROD124
      SAT(IT) = SA                                                      AEROD125
      CLAMAR(IT) = SCT(IT) / (SAT(IT) * CREF)                           AEROD126
      GO TO 50                                                          AEROD127
 80   CONTINUE                                                          AEROD128
      IF (IXX.EQ.1) GO TO 100                                           AEROD129
      IF (IPLAN.EQ.1) GO TO 90                                          AEROD130
      SCT(IT) = SC                                                      AEROD131
      SAT(IT) = SA                                                      AEROD132
      CLAMAR(IT) = (SCT(IT) - SCT(IT-1)) / ((SAT(IT) - SAT(IT-1)) *CREF)AEROD133
      GO TO 100                                                         AEROD134
 90   CLAMAR(1)=SC/(SA*CREF)                                            AEROD135
 100  CONTINUE                                                          AEROD136
      IF (TWST.EQ.0..AND.IXX.EQ.1) GO TO 110                            AEROD137
      YCP(IXX)=SB/(SA*BOT)                                              AEROD138
      AC(IXX)=SC/(SA*CREF)                                              AEROD139
      GO TO 120                                                         AEROD140
 110  YCP(1)=AC(1)=0.                                                   AEROD141
 120  CONTINUE                                                          AEROD142
      CMCL=AC(2)                                                        AEROD143
      CMO=(AC(1)-AC(2))*CLT                                             AEROD144
C                                                                       AEROD145
C                                PART 3 - SECTION 2                     AEROD146
C             COMPUTE OTHER-  AND PRINT ALL FINAL-  OUTPUT DATA HERE    AEROD147
C                                                                       AEROD148
      DO 140 IXX=1,2                                                    AEROD149
      JN=0                                                              AEROD150
      DO 140 JSSW=1,NSSW                                                AEROD151
      CH(IXX,JSSW)=0                                                    AEROD152
      NSCW=TBLSCW(JSSW)                                                 AEROD153
      DO 130 JSCW=1,NSCW                                                AEROD154
      JN=JN+1                                                           AEROD155
      CH(IXX,JSSW)=(-2.0)*(PV(JN)-PN(JN))*BETA+CH(IXX,JSSW)             AEROD156
 130  CONTINUE                                                          AEROD157
      CCAV(IXX,JSSW)=CH(IXX,JSSW)/CAVE                                  AEROD158
      CLCL(IXX,JSSW)=SLOAD(IXX,JSSW)/CCAV(IXX,JSSW)                     AEROD159
 140  CONTINUE                                                          AEROD160
      CLD=CLDES                                                         AEROD161
      IF (CLDES.EQ.11) CLD=1.                                           AEROD162
      DO 150 I=1,M                                                      AEROD163
      CP(I)=(CLCC(I,1)+CLCC(I,2)*(CLD-CLT)/CLNT)*CAVE/(2.*(PN(I)-PV(I))*AEROD164
     1BETA)                                                             AEROD165
 150  CONTINUE                                                          AEROD166
      WRITE (6,240) CONFIG                                              AEROD167
      IF (PTEST.NE.0.) WRITE (6,350)                                    AEROD168
      IF (QTEST.NE.0.) WRITE (6,330)                                    AEROD169
      IF (PTEST.EQ.0..AND.QTEST.EQ.0.) WRITE (6,340)                    AEROD170
```

```
      WRITE (6,360) CLD
      HEAD=8HDESIRED
      IF (CLDES.EQ.11.) HEAD=8H
      IEND=11
      IF (CLDES.NE.11.) IEND=1
      DO 190 IUTK=1,IEND
      IF (IEND.EQ.11) CLDES=(FLOAT(IUTK)-1.)/10.
      IF (CLDES.EQ.0.) CLDES=-.1
      NR=0
      DO 160 NV=1,NSSW
      NSCW=TBLSCW(NV)
      NP=NR+1
      NR=NR+NSCW
      PHIPR=ATAN(PHI(NV))*RAD
      SLOAD(3,NV)=0.
      IF (IUTK .NE. 1) GO TO 155
      NVS = 1
      DO 154 IT = 1,IPLAN
      JJ = IT - 1
      IF(JJ.EQ.0) GOTO 153
      NVS = NVS + NSSWSV(JJ)
  153 IF(NV.NE.NVS) GOTO 154
      WRITE(6,230) NUMBER(IT)
      GOTO 155
  154 CONTINUE
  155 CONTINUE
      DO 160 I=NP,NR
      IF (IUTK.GT.1) GO TO 160
      PNPR=PN(I)*BETA
      PVPR=PV(I)*BETA
      PSIPR=ATAN(BETA*TAN(PSI(I)))*RAD
      WRITE (6,370) PNPR,PVPR,Q(I),ZH(NV),S(I),PSIPR,PHIPR,ALP(I),CP(I)
160   SLOAD(3,NV)=SLOAD(3,NV)+CLCC(I,2)*CLDES/CLNT+CLCC(I,1)-CLCC(I,2)*C
     1LT/CLNT
      IF (IUTK.GT.1) GO TO 170
      WRITE (6,270)
      WRITE (6,280) CREF,CAVE,STRUE,SREF,BOT,AR,ARTRUE,MACH
170   CONTINUE
C
C
      IF (PTEST.NE.0.) WRITE (6,380) CLP
      IF (PTEST.NE.0.) GO TO 220
C
C     COMPUTE CMQ,CLQ
C
      CMQ=2.0*CMCL*CLNT/(0.08725*CREF)
      CLQ=2.0*CLNT/(0.08725*CREF)
      IF (QTEST.NE.0.) WRITE (6,390) CMQ,CLQ
      IF (QTEST.NE.0.) GO TO 220
```

```
C
C        COMPUTE INDUCED DRAG FOR FLAT WING-BODY WITH NO DIHEDRAL
C
      NSV = 1
      MTOT = 1
      DO 180 IT = 1,KBOT
      MTOT = MTOT + MSV(IT)
      NSV = NSV + NSSWSV(IT)
  180 CONTINUE
      CALL CDICLS(AR,ARTRUE,NSSWSV(KBOT),MTOT,NSV,CDI,CDIT)
      CLAPD=CLA(2)/57.29578
      ALPO=-(CLT/CLA(2))*57.29578
      ALPD=CLDES/CLAPD+ALPO
      ALPW=1./CLAPD
      CLWB=CLWING*ALPD/57.29578+CLWNGT
      CDIWB=CDI/(CLWB*CLWB)
      IF (IUTK.EQ.1) WRITE (6,250) HEAD,CDIT
 190  WRITE (6,260) CLDES,ALPD,CLWB,CDI,CDIWB
      WRITE (6,290) CLA(2),CLAPD,CLT,ALPO,YCP(2),CMCL,CMO
      YYCP(1) = YCP(2)
      IF(IPLAN.EQ.1) GOTO 194
      DO 193 IT=1,IPLAN
      IF(IT.GT.1) GOTO 191
      CLTWST  = CLWNG(1)
      CLALPHA = CLWIN(1)
      YYCP(1) = CROLL(1) / CLALPHA
      GOTO 192
  191 CLTWST = CLWNG(IT) - CLWNG(IT-1)
      CLALPHA = CLWIN(IT) - CLWIN(IT-1)
      YYCP(IT) = (CROLL(IT) - CROLL(IT-1)) / CLALPHA
  192 CLAPDIT = CLALPHA / 57.29578
      ALPOIT = - ( CLTWST / CLALPHA ) * 57.29578
      WRITE(6,400) NUMBER(IT), CLALPHA, CLAPDIT, CLTWST,
     $              ALPOIT, YYCP(IT)
  193 CONTINUE
  194 CONTINUE
      WRITE (6,300) CLT
      NR=J=0
      DO 210 NV=1,NSSW
      BCLCC=BADLAE=BASLD=0.
      NSCW=TBLSCW(NV)
      NP=NR+1
      NR=NR+NSCW
      DO 200 I=NP,NR
      ADLAE=CLCC(I,2)*CLT/CLNT
      BSLD=CLCC(I,1)-ADLAE
      BCLCC=BCLCC+CLCC(I,1)
      BADLAE=BADLAE+ADLAE
      BASLD=BASLD+BSLD
```

```
200    CONTINUE                                                            AERO0269
       SLDT(NV)=(SMOAD(1,NV)+SMOAD(2,NV)*(CLDES-CLT)/CLNT)/(SMLD(1,NV)+SMAERO0270
      1LD(2,NV)*(CLDES-CLT)/CLNT)                                          AERO0271
       J=J+NSCW                                                            AERO0272
       YQ=Q(J)/BOT                                                         AERO0273
       NVS = 1                                                             AERO0274
       DO 205 IT = 1,IPLAN                                                 AERO0275
       JJ = IT - 1                                                         AERO0276
       IF(JJ.EQ.0) GOTO 204                                                AERO0277
       NVS = NVS + NSSWSV(JJ)                                              AERO0278
   204 IF(NV.NE.NVS) GOTO 205                                              AERO0279
       WRITE(6,310) NUMBER(IT)                                             AERO0280
       GOTO 206                                                            AERO0281
   205 CONTINUE                                                            AERO0282
   206 CONTINUE                                                            AERO0283
210    WRITE (6,320) NV,YQ,SLOAD(2,NV),CLCL(2,NV),CCAV(2,NV),BCLCC,BADLAEAERO0284
      1,BASLD,SLOAD(3,NV),SLDT(NV)                                         AERO0285
220    CONTINUE                                                            AERO0286
       CLDES=CLSAVE                                                        AERO0287
       IF(ITIPCOD.EQ.2) CALL FLOWFL                                        AERO0288
C                                                                          AERO0289
C                                                                          AERO0290
C                                                                          AERO0291
   230 FORMAT(/12X,A6,39H PLANFORM HORSESHOE VORTEX DESCRIPTIONS/)         AERO0292
   240 FORMAT(1H1,///,55X,16HAERODYNAMIC DATA ,///,54X,                    AERO0293
      $ 16HCONFIGURATION : ,2A10)                                          AERO0294
250    FORMAT (1H1,18X,22HCOMPLETE CONFIGURATION,31X,25HWING-BODY CHARACTAERO0295
      1ERISTICS/64X,4HLIFT,9X,33HINDUCED DRAG (FAR FIELD SOLUTION)//16XA8AERO0296
      2,21H CL     COMPUTED ALPHA,19X,6HCL(WB),7X,13HCDI AT CL(WB),4X,15HCAERO0297
      $DI/(CL(WB)**2),/,88X,16H(1/PI*AR REF) = ,F8.5,1H))                  AERO0298
260    FORMAT (11X,2F15.5,15X,3F15.5)                                      AERO0299
270    FORMAT (////4X,11H REF. CHORD,6X,25HC AVERAGE      TRUE AREA  ,2X,1AERO0300
      14HREFERENCE AREA,9X,3HB/2,8X,7HREF. AR,8X,7HTRUE AR,4X,11HMACH NUMAERO0301
      2BER/)                                                               AERO0302
280    FORMAT (8F15.5)                                                     AERO0303
290    FORMAT (///47X,38HCOMPLETE CONFIGURATION CHARACTERISTICS//36X,8HCLAERO0304
      1 ALPHA,8X,53HCL(TWIST)  ALPHA AT CL=0     Y CP       CM/CL       CMOAERO0305
      2/27X,23HPER RADIAN    PER DEGREE/24X,7F12.5)                        AERO0306
300    FORMAT (//25X,18HADDITIONAL LOADING/24X,24HWITH CL BASED ON S(TRUEAERO0307
      1)71X,11H-AT CL DES-/67X,34HLOAD DUE  ADD. LOAD AT  BASIC LOAD3X,27AERO0308
      2HSPAN LOAD AT  X LOCATION OF/8H STATION6X,5H 2Y/B9X,9H SL COEF ,4XAERO0309
      3,8HCL RATIO4X,7HC RATIO,7X,14HTO TWIST   CL=,F9.5,3X,7HAT CL=05X,2AERO0310
      46HDESIRED CL   LOCAL CENT PR/)                                      AERO0311
   310 FORMAT(/,47X,A6,32H PLANFORM SPAN LOAD DISTRIBUTION ,/)             AERO0312
320    FORMAT (4X,I4,F12.5,5X,3F12.5,3X,3F12.5,3X,2F12.5)                  AERO0313
330    FORMAT (/54X,24HCMQ AND CLQ ARE COMPUTED//)                         AERO0314
340    FORMAT (/38X,57HSTATIC LONGITUDINAL AERODYNAMIC COEFFICIENTS ARE CAERO0315
      1OMPUTED//)                                                          AERO0316
350    FORMAT (/59X,15HCLP IS COMPUTED//)                                  AERO0317
```

```
360   FORMAT (/20X,1HX,11X,1HX,11X,1HY,11X,1HZ,12X,1HS,5X,9HC/4 SWEEP,4XAEROD318
     1,8HDIHEDRAL,2X,11HLOCAL ALPHA,2X,19HDELTA CP AT DESIRED/19X,3HC/4,AEROD319
     29X,4H3C/4,42X,5HANGLE,7X,5HANGLE,4X,10HIN RADIANS,4X,4HCL =,F10.5/AEROD320
     3)                                                                  AEROD321
370   FORMAT (12X,9F12.5)                                                AEROD322
  380 FORMAT(//////////,56X,4HCLP=,F12.4,////)                           AEROD323
  390 FORMAT(//////////,42X,4HCMQ=,F12.5,10X,4HCLQ=,F12.5,////)          AEROD324
 400  FORMAT(/, 1X, A10, 8HPLANFORM, 5X, 5F12.5)                         AEROD325
      END                                                                AEROD326
      SUBROUTINE FLOWFL                                                  FLOWFL 2
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                 ALL     2
     $    STA(4), TBLSCW(100), YYCP(4),                                  ALL     3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),          ALL     4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                         ALL     5
C                                                                        ALL     6
      COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4),   MAINONE2
     $  XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),        MAINONE3
     $  XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),     MAINONE4
     $  ZZ(25,4), ITIPCOD, ICAMTST                                       MAINONE5
C                                                                        MAINONE6
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,     ONETHRE2
     $   ARTRUE, RTCOHT(4), CONFIG(2), NSSWSV(4),                        ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                                ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),       ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                           ONETHRE6
     $  YINNER(4), YOUTER(4)                                             ONETHRE7
C                                                                        ONETHRE8
      INTEGER CONFIG                                                     ONETHRE9
C                                                                        ONETHR10
      COMMON /TOTHREE/ CIR(400,2)                                        TOTHRE 2
C                                                                        TOTHRE 3
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD                THREFOR2
C                                                                        THREFOR3
C                                                                        FLOWFL 8
      COMMON / HEAP / BREAKPT(75,9), INDEXL, INDEXR, NUMSTA, NUMSTOT     HEAP    2
C                                                                        HEAP    3
      DIMENSION YFL(75), XCVLE(75), XCVTE(75)                            HEAP    4
      EQUIVALENCE (BREAKPT(1,1),YFL(1)),                                 HEAP    5
     $              (BREAKPT(1,2),XCVLE(1)),                             HEAP    6
     $              (BREAKPT(1,3),XCVTE(1))                              HEAP    7
C                                                                        HEAP    8
C      COL1     COL2  COL3  COL4  COL5  COL6  COL7  COL8  COL9           HEAP    9
C      YFL      XLE1  XTE1  XLE2  XTE2  XLE3  XTE3  XLE4  XTE4           HEAP   10
C                                                                        HEAP   11
C                                                                        FLOWFL10
      COMMON  / INSUB23 /  APSI,APHI,XXX,YYY,ZZZ,SNN,TOLC                FLOWFL11
      LOGICAL SKIPIT                                                     FLOWFL12
      DIMENSION GAMP(400,4), YA(2), FV(2), FW(2), FU(2)                  FLOWFL13
      LOGICAL OFFPLAN(75)                                                FLOWFL14
```

```
C                                                                       FLOWFL15
      DIMENSION MMM(4),NUMBER(4)                                        FLOWFL16
      DIMENSION INDEXES(400,2), YFLOW(75), XFLOW(75)                    FLOWFL17
C                                                                       FLOWFL18
      REAL MACH                                                         FLOWFL19
      DATA NUMBER/5HFIRST,6HSECOND,5HTHIRD,6HFOURTH/                    FLOWFL20
C                                                                       FLOWFL21
      NUMSTA = 75                                                       FLOWFL22
      NUMSTOT = 75                                                      FLOWFL23
      DO 15 J = 1,9                                                     FLOWFL24
        DO 10 I = 1,NUMSTA                                              FLOWFL25
          BREAKPT(I,J) = 0.                                             FLOWFL26
   10   CONTINUE                                                        FLOWFL27
   15 CONTINUE                                                          FLOWFL28
C                                                                       FLOWFL29
C                                                                       FLOWFL30
      TOLC = (BOT * 15.E - 05) ** 2                                     FLOWFL31
C                                                                       FLOWFL32
      WRITE (6,630)                                                     FLOWFL33
      WRITE (6,620)                                                     FLOWFL34
C                                                                       FLOWFL35
      PI = 4. * ATAN(1.)                                                FLOWFL36
      RAD = 180. / PI                                                   FLOWFL37
      CONST = 4. * PI                                                   FLOWFL38
C                                                                       FLOWFL39
C                                                                       FLOWFL40
C             ALPW AND ALPO USED HEREIN IN RADIANS                      FLOWFL41
C                                                                       FLOWFL42
C                                                                       FLOWFL43
      ALPW = 1. / CLNT                                                  FLOWFL44
      ALPO = -(CLT / CLNT)                                              FLOWFL45
      ALPW = ALPW * CLDES                                               FLOWFL46
      ALPD = ALPW + ALPO                                                FLOWFL47
C                                                                       FLOWFL48
C                                                                       FLOWFL49
C             GAMP(J,1) IS THE BASIC LOAD INITIALLY                     FLOWFL50
C             GAMP(J,2) IS THE ADDITIONAL LOAD AT CL = 1.               FLOWFL51
C             GAMP(J,3) IS THE TOTAL LOAD AT CLDES                      FLOWFL52
C             GAMP(J,4) IS THE ADDITIONAL LOAD AT CLDES                 FLOWFL53
C                                                                       FLOWFL54
      DO 20 I = 1,M                                                     FLOWFL55
      GAMP(I,1) = CIR(I,1) + CIR(I,2) * ( - CLT / CLNT)                 FLOWFL56
      GAMP(I,2) = CIR(I,2) / CLNT                                       FLOWFL57
   20 CONTINUE                                                          FLOWFL58
      DO 30 K = 1,M                                                     FLOWFL59
      GAMP(K,4) = GAMP(K,2) * CLDES                                     FLOWFL60
      GAMP(K,3) = GAMP(K,1) + GAMP(K,4)                                 FLOWFL61
   30 CONTINUE                                                          FLOWFL62
C                                                                       FLOWFL63
```

```
      WRITE(6,640) CONFIG                                               FLOWFL64
      WRITE (6,670) CLDES,CLDES                                         FLOWFL65
C                                                                       FLOWFL66
      NN = 1                                                            FLOWFL67
      LMA = TBLSCW(1)                                                   FLOWFL68
      MMM(1) = 1                                                        FLOWFL69
      MMM(2) = 1 + MSV(1)                                               FLOWFL70
      MMM(3) = 1 + MSV(1) + MSV(2)                                      FLOWFL71
      MMM(4) = 1 + MSV(1) + MSV(2) + MSV(3)                             FLOWFL72
      DO 60 I = 1,NSSW                                                  FLOWFL73
      J = NN                                                            FLOWFL74
      J1 = J + 1                                                        FLOWFL75
C                                                                       FLOWFL76
      DO 40 IT=1,IPLAN                                                  FLOWFL77
      IF(J.EQ.MMM(IT)) WRITE(6,720) NUMBER(IT)                          FLOWFL78
   40 CONTINUE                                                          FLOWFL79
      WRITE (6,680) I,Q(J),GAMP(J,1),GAMP(J,2),GAMP(J,3),GAMP(J,4)      FLOWFL80
C                                                                       FLOWFL81
      IF(TBLSCW(I).EQ.1) GO TO 50                                       FLOWFL82
C                                                                       FLOWFL83
      WRITE (6,690) (GAMP(L,1),GAMP(L,2),GAMP(L,3),GAMP(L,4),L = J1,LMA)FLOWFL84
C                                                                       FLOWFL85
      IF (I .EQ. NSSW) GO TO 60                                         FLOWFL86
      LMA=TBLSCW(I)+LMA                                                 FLOWFL87
   50 NN=TBLSCW(I)+NN                                                   FLOWFL88
   60 CONTINUE                                                          FLOWFL89
C                                                                       FLOWFL90
      LL = 1                                                            FLOWFL91
      MM = 0                                                            FLOWFL92
      K = 0                                                             FLOWFL93
C                                                                       FLOWFL94
      DO 80 ITT = 1,IPLAN                                               FLOWFL95
        JRANGE = NSSWSV(ITT)                                            FLOWFL96
        DO 70 J = 1,JRANGE                                              FLOWFL97
          K = K + 1                                                     FLOWFL98
          MM = MM + TBLSCW(K)                                           FLOWFL99
          INDEXES(K,2) = MM                                             FLOWF100
          XCVLE(K) = .5 * BETA * (3. * PN(LL) - PV(LL))                 FLOWF101
          XCVTE(K) = .5 * BETA * (3. * PV(MM) - PN(MM))                 FLOWF102
          INDEXES(K,1) = LL                                             FLOWF103
          YFL(K) = Q(LL)                                                FLOWF104
          LL = LL + TBLSCW(K)                                           FLOWF105
   70   CONTINUE                                                        FLOWF106
   80 CONTINUE                                                          FLOWF107
C                                                                       FLOWF108
C     SORT THE VALUES IN -YFL- AND REMOVE THE DUPLICATES. NOTE          FLOWF109
C     FOR DUPLICATES, THEIR XLE AND XTE VALUES ARE STORED IN THE        FLOWF110
C     NEXT AVAILABLE COLUMNS OF THE -BREAKPT- MATRIX                    FLOWF111
C                                                                       FLOWF112
```

```
      NUMSTA = K                                                        FLOWF113
      CALL HEAPSRT                                                      FLOWF114
      NUMPNTS = NUMSTA                                                  FLOWF115
      REMAIND = NUMSTOT - NUMPNTS                                       FLOWF116
      DELTAY = 2.*BOT/REMAIND                                           FLOWF117
C                                                                       FLOWF118
C                                                                       FLOWF119
C                                                                       FLOWF120
C             READ IN TOTAL NUMBER OF FIELD LINES                       FLOWF121
C                                                                       FLOWF122
      READ(5,610) TOTFL                                                 FLOWF123
C                                                                       FLOWF124
C                                                                       FLOWF125
      NTOTFL = TOTFL                                                    FLOWF126
      DO 160 NTOT = 1,NTOTFL                                            FLOWF127
      RSTA = NTOT                                                       FLOWF128
      SKIPIT = .FALSE.                                                  FLOWF129
C                                                                       FLOWF130
C                                                                       FLOWF131
C             F I E L D   L I N E   D E S C R I P T I O N               FLOWF132
C                                                                       FLOWF133
C                                                                       FLOWF134
C             XDOWN IS THE X LOCATION AT THE PLANE OF SYMMERTRY         FLOWF135
C                                                                       FLOWF136
C             SWEP IS THE SWEEP ANGLE IN DEGREES                        FLOWF137
C                                                                       FLOWF138
C             ZREF IS THE Z LOCATION AT THE PLANE OF SYMMERTRY          FLOWF139
C                                                                       FLOWF140
C             DIHED IS THE DIHEDRAL ANGLE IN DEGREES                    FLOWF141
C                                                                       FLOWF142
C                                                                       FLOWF143
      READ (5,610) XDOWN,SWEP,ZREF,DIHED                                FLOWF144
      XDOWN=XDOWN-XLOCTN                                                FLOWF145
C                                                                       FLOWF146
C                                                                       FLOWF147
      TANFL = TAN(SWEP / RAD)                                           FLOWF148
      TDIHED = TAN(DIHED / RAD)                                         FLOWF149
      WRITE (6,650)                                                     FLOWF150
      ALPWW = ALPD * RAD                                                FLOWF151
C                                                                       FLOWF152
      WRITE(6,660) RSTA,XDOWN,SWEP,ZREF,DIHED,ALPWW,CLDES               FLOWF153
      WRITE (6,700)                                                     FLOWF154
C                                                                       FLOWF155
C                                                                       FLOWF156
C          NOTE THAT THE PLANFORM OF MAXIMUM                            FLOWF157
C          SEMI-SPAN MUST EXTEND TO THE PLANE                           FLOWF158
C          OF SYMMETRY.                                                 FLOWF159
      DO 110 I = 1,NUMPNTS                                              FLOWF160
      OFFPLAN(I) = .TRUE.                                               FLOWF161
```

```
      YFLOW(I) = YFL(I)                                                 FLOWF162
      XFLOW(I) = YFL(I) * TANFL + XDOWN                                 FLOWF163
      DO 100 J=1,4                                                      FLOWF164
        JT2 = J * 2                                                     FLOWF165
        JT2P1 = J * 2 + 1                                               FLOWF166
        XLE = BREAKPT(I,JT2)                                            FLOWF167
        XTE = BREAKPT(I,JT2P1)                                          FLOWF168
        IF((ABS(XLE) .LT. ABS(XFLOW(I))).AND.                           FLOWF169
     $     (ABS(XFLOW(I)).LT. ABS(XTE))) OFFPLAN(I) = .FALSE.           FLOWF170
  100   CONTINUE                                                        FLOWF171
  110 CONTINUE                                                          FLOWF172
  115 J = 0                                                             FLOWF173
      K = NUMPNTS + 1                                                   FLOWF174
      DO 120 I = K,NUMSTOT                                              FLOWF175
        J = J + 1                                                       FLOWF176
        OFFPLAN(I) = .TRUE.                                             FLOWF177
        YFLOW(I) = -BOT - J*DELTAY                                      FLOWF178
        XFLOW(I) = YFLOW(I) * TANFL + XDOWN                             FLOWF179
  120 CONTINUE                                                          FLOWF180
C                                                                       FLOWF181
C                                                                       FLOWF182
C                                                                       FLOWF183
C                                                                       FLOWF184
C                                                                       FLOWF185
C                                                                       FLOWF186
      DO 150 II = 1,NUMSTOT                                             FLOWF187
      IF (OFFPLAN(II)) GO TO 125                                        FLOWF188
      ZFL = ZREF + YFLOW(II) * TDIHED                                   FLOWF189
      WRITE(6,730) II, XFLOW(II), YFLOW(II), ZFL                        FLOWF190
      SKIPIT = .TRUE.                                                   FLOWF191
      GO TO 150                                                         FLOWF192
  125 WOU = 0.                                                          FLOWF193
      ZFL = ZREF + YFLOW(II) * TDIHED                                   FLOWF194
      UOU = 0                                                           FLOWF195
      VOU = 0                                                           FLOWF196
      DWOUDA = 0                                                        FLOWF197
      DUOUDA = 0                                                        FLOWF198
      XFLOW(II) = XFLOW(II) / BETA                                      FLOWF199
      IZ = 1                                                            FLOWF200
      NNN = TBLSCW(IZ)                                                  FLOWF201
      DO 140 NN = 1,M                                                   FLOWF202
      SNN = S(NN)                                                       FLOWF203
      XXX = XFLOW(II) - PN(NN)                                          FLOWF204
      ZZZ = ZFL - ZH(IZ)                                                FLOWF205
      APHI = ATAN(PHI(IZ))                                              FLOWF206
      APSI = PSI(NN)                                                    FLOWF207
      YA(1) = YFLOW(II) - Q(NN)                                         FLOWF208
      YA(2) = YFLOW(II) + Q(NN)                                         FLOWF209
      DO 130 I = 1,2                                                    FLOWF210
```

```
      YYY = YA(I)                                                       FLOWF211
      CALL INFSUB (BOT,FV(I),FW(I),FU(I))                               FLOWF212
      APHI = -APHI                                                      FLOWF213
      APSI = -APSI                                                      FLOWF214
  130 CONTINUE                                                          FLOWF215
      WOU = WOU + (FW(1) + FW(2)) * GAMP(NN,3) / CONST                  FLOWF216
      VOU = VOU + (FV(1) + FV(2)) * GAMP(NN,3) / CONST                  FLOWF217
      UOU = UOU + (FU(1) + FU(2)) * GAMP(NN,3) / CONST                  FLOWF218
      DWOUDA = DWOUDA + (FW(1) + FW(2)) * GAMP(NN,4) / (CONST * ALPW)   FLOWF219
      DUOUDA = DUOUDA + (FU(1) + FU(2)) * GAMP(NN,4) / (CONST * ALPW)   FLOWF220
      IF (NN .LT. NNN .OR. NN .EQ. M) GO TO 140                         FLOWF221
      IZ = IZ + 1                                                       FLOWF222
      NNN = NNN + TBLSCW(IZ)                                            FLOWF223
  140 CONTINUE                                                          FLOWF224
      CA = COS(ALPD)                                                    FLOWF225
      SA = SIN(ALPD)                                                    FLOWF226
      EPSILN = (ATAN((WOU * CA - UOU * SA) / (1. - WOU * SA - UOU * CA))FLOWF227
     1) * RAD                                                           FLOWF228
      DEPDAL = (WOU ** 2 + UOU ** 2 - (WOU + DUOUDA) * SA + (DWOUDA - UOFLOWF229
     1U) * CA + WOU * DUOUDA - UOU * DWOUDA) / (1. - 2. * (WOU * SA + UOFLOWF230
     2U * CA) + UOU ** 2 + WOU ** 2)                                    FLOWF231
      QRATI = 1. - 2. * (UOU * CA + WOU * SA) + UOU ** 2 + VOU ** 2 + WOFLOWF232
     1U ** 2                                                            FLOWF233
      SMACH = MACH * MACH                                               FLOWF234
      QRATIO = QRATI * (1. + .2 * SMACH * (1. - QRATI)) ** 2.5          FLOWF235
      DWNWH = ATAN(WOU) * RAD                                           FLOWF236
      SIGMA = ATAN(VOU) * RAD                                           FLOWF237
      XFLOW(II) = XFLOW(II) * BETA                                      FLOWF238
C                                                                       FLOWF239
      WRITE(6,710)II,XFLOW(II),YFLOW(II),ZFL,WOU,VOU,UOU,DWNWH,EPSILN   FLOWF240
     1,DEPDAL,QRATIO,SIGMA                                              FLOWF241
C                                                                       FLOWF242
  150 CONTINUE                                                          FLOWF243
      IF (SKIPIT) WRITE(6,740)                                          FLOWF244
  160 CONTINUE                                                          FLOWF245
      RETURN                                                            FLOWF246
C                                                                       FLOWF247
  610 FORMAT (8F10.6)                                                   FLOWF248
  620 FORMAT (1H1)                                                      FLOWF249
  630 FORMAT (1H1/////////////////////////48X,43HF L O W   F I E L D   I FLOWF250
     1N F O R M A T I O N)                                              FLOWF251
  640 FORMAT(//,58X,16HCONFIGURATION : ,2A10)                           FLOWF252
  650 FORMAT (1H1///60X,15HFIELD LINE DATA//20X,10HFIELD LINE,4X,1HX,5X,FLOWF253
     113HSWEEP,DEGREES,6X,1HZ,4X,16HDIHEDRAL,DEGREES,2X,13HALPHA,DEGREESFLOWF254
     2,3X,7HWING CL)                                                    FLOWF255
  660 FORMAT (15X,F12.0,4F12.5,6X,2F12.5)                               FLOWF256
  670 FORMAT (30X,7HSTATION,4X,3H Y ,26X,7HGAMMA/U/,53X,41HBASIC  ADDITIFLOWF257
     1ONAL        TOTAL  ADDITIONAL/,48X,31HAT CL = 0.  AT CL = 1.0 AT CLFLOWF258
     2 =,F5.2,2X,7HAT CL =,F5.2)                                        FLOWF259
```

```
  680 FORMAT (29X,I5,5F12.5)                                            FLOWF260
  690 FORMAT (46X,4F12.5)                                               FLOWF261
  700 FORMAT (//,80X,8HDOWNWASH,/,1X,5HFIELD,6X,1HX,11X,1HY,11X,1HZ,11X,FLOWF262
     13HW/U,9X,3HV/U,9X,3HU/U,6X,6HANGLE,,5X,,8HEPSILON,,1X,10HD(EPSILONFLOWF263
     2),2X,8HQ(LOCAL),5X,6HSIGMA,,/,1X,5HPOINT,74X,7HDEGREES,5X,7HDEGREEFLOWF264
     3S,3X,8HD(ALPHA),4X,6HQ(INF),5X,7HDEGREES,/)                       FLOWF265
  710 FORMAT(1X,I3,1H.,6F12.5,5F11.5)                                   FLOWF266
 720  FORMAT(//42X,32HINPUT VALUES OF CIRCULATION FOR ,A10,             FLOWF267
     * 8HPLANFORM//)                                                    FLOWF268
  730 FORMAT(1X,I3,1H.,3F12.5,12X,5H***** )                             FLOWF269
  740 FORMAT(1H0,5X,28H***** - COMPUTATION OMITTED ,                    FLOWF270
     $ 41HDUE TO THE POINT PROJECTION LYING WITHIN ,                    FLOWF271
     $ 26HTHE CONFIGURATION BOUNDARY ,//)                               FLOWF272
C                                                                       FLOWF273
      END                                                               FLOWF274
      SUBROUTINE HEAPSRT                                                HEAPSRT2
C                                                                       HEAPSRT3
C        THIS ROUTINE PERFORMS A HEAPSORT ON THE VALUES IN              HEAPSRT4
C                                                                       HEAPSRT5
C     THE YFL ARRAY, AND MOVES THE CORRESPONDING XLE AND XTE            HEAPSRT6
C     VALUES WHEN A SWAP IS MADE. THIS ALGORITHM IS DESCRIBED           HEAPSRT7
C     IN N. WIRTH'S "ALGORITHMS + DATA STRUCTURES = PROGRAMS"           HEAPSRT8
C     BOOK, 1976, PRENTICE-HALL,INC.                                    HEAPSRT9
C                                                                       HEAPSR10
C                                                                       HEAPSR11
      COMMON / HEAP / BREAKPT(75,9), INDEXL, INDEXR, NUMSTA, NUMSTOT    HEAP   2
C                                                                       HEAP   3
      DIMENSION YFL(75), XCVLE(75), XCVTE(75)                           HEAP   4
      EQUIVALENCE (BREAKPT(1,1),YFL(1)),                                HEAP   5
     $            (BREAKPT(1,2),XCVLE(1)),                              HEAP   6
     $            (BREAKPT(1,3),XCVTE(1))                               HEAP   7
C                                                                       HEAP   8
C     COL1      COL2  COL3  COL4  COL5  COL6  COL7  COL8  COL9          HEAP   9
C     YFL       XLE1  XTE1  XLE2  XTE2  XLE3  XTE3  XLE4  XTE4          HEAP  10
C                                                                       HEAP  11
C                                                                       HEAPSR13
      N = NUMSTA                                                        HEAPSR14
      INDEXL = (N / 2) + 1                                              HEAPSR15
      INDEXR = N                                                        HEAPSR16
   10 CONTINUE                                                          HEAPSR17
      IF (INDEXL .LE. 1) GO TO 20                                       HEAPSR18
        INDEXL = INDEXL - 1                                             HEAPSR19
        CALL SIFT                                                       HEAPSR20
        GO TO 10                                                        HEAPSR21
C                                                                       HEAPSR22
   20 CONTINUE                                                          HEAPSR23
      IF ( INDEXR .LE. 1) GO TO 30                                      HEAPSR24
      Y1 = YFL(1)                                                       HEAPSR25
      X1 = XCVLE(1)                                                     HEAPSR26
```

```
      X2 = XCVTE(1)                                                     HEAPSR27
      YFL(1) = YFL(INDEXR)                                              HEAPSR28
      XCVLE(1) = XCVLE(INDEXR)                                          HEAPSR29
      XCVTE(1) = XCVTE(INDEXR)                                          HEAPSR30
      YFL(INDEXR) = Y1                                                  HEAPSR31
      XCVLE(INDEXR) = X1                                                HEAPSR32
      XCVTE(INDEXR) = X2                                                HEAPSR33
        INDEXR = INDEXR - 1                                             HEAPSR34
        CALL SIFT                                                       HEAPSR35
        GO TO 20                                                        HEAPSR36
   30 CONTINUE                                                          HEAPSR37
C     REMOVE THE DUPLICATE -YFL- VALUES BY COMBINING THEIR              HEAPSR38
C     XLE AND XTE VALUES IN -BREAKPT-                                   HEAPSR39
C                                                                       HEAPSR40
      N = 1                                                             HEAPSR41
      KK = 4                                                            HEAPSR42
      TEMP1 = YFL(1)                                                    HEAPSR43
      DO 50 J=2,NUMSTA                                                  HEAPSR44
        IF(TEMP1 .EQ. YFL(J)) GO TO 40                                  HEAPSR45
        KK = 4                                                          HEAPSR46
        N = N + 1                                                       HEAPSR47
        YFL(N) = YFL(J)                                                 HEAPSR48
        XCVLE(N) = XCVLE(J)                                             HEAPSR49
        XCVTE(N) = XCVTE(J)                                             HEAPSR50
        TEMP1 = YFL(J)                                                  HEAPSR51
        GO TO 50                                                        HEAPSR52
   40   CONTINUE                                                        HEAPSR53
        BREAKPT(N,KK) = XCVLE(J)                                        HEAPSR54
        BREAKPT(N,KK+1) = XCVTE(J)                                      HEAPSR55
        KK = KK + 2                                                     HEAPSR56
   50 CONTINUE                                                          HEAPSR57
      NUMSTA = N                                                        HEAPSR58
      RETURN                                                            HEAPSR59
      END                                                               HEAPSR60
      SUBROUTINE SIFT                                                   SIFT   2
      COMMON / HEAP / BREAKPT(75,9), INDEXL, INDEXR, NUMSTA, NUMSTOT    HEAP   2
C                                                                       HEAP   3
      DIMENSION YFL(75), XCVLE(75), XCVTE(75)                           HEAP   4
      EQUIVALENCE (BREAKPT(1,1),YFL(1)),                                HEAP   5
     $            (BREAKPT(1,2),XCVLE(1)),                              HEAP   6
     $            (BREAKPT(1,3),XCVTE(1))                               HEAP   7
C                                                                       HEAP   8
C      COL1     COL2  COL3  COL4  COL5  COL6  COL7  COL8  COL9          HEAP   9
C      YFL      XLE1  XTE1  XLE2  XTE2  XLE3  XTE3  XLE4  XTE4          HEAP  10
C                                                                       HEAP  11
      INDEXI = INDEXL                                                   SIFT   4
      INDEXJ = 2 * INDEXI                                               SIFT   5
      Y1 = YFL(INDEXI)                                                  SIFT   6
      X1 = XCVLE(INDEXI)                                                SIFT   7
```

```
      X2 = XCVTE(INDEXI)                                                SIFT    8
   10 CONTINUE                                                          SIFT    9
      IF (INDEXJ .GT. INDEXR) GO TO 30                                  SIFT   10
        IF (INDEXJ .EQ. INDEXR) GO TO 20                                SIFT   11
          JP1 = INDEXJ + 1                                              SIFT   12
        ABSYJ = ABS(YFL(INDEXJ))                                        SIFT   13
        ABSYJP1 = ABS(YFL(JP1))                                         SIFT   14
          IF (ABSYJ .LT. ABSYJP1) INDEXJ = JP1                          SIFT   15
   20 CONTINUE                                                          SIFT   16
      IF(ABS(Y1) .GE. ABS(YFL(INDEXJ))) GO TO 30                        SIFT   17
      YFL(INDEXI) = YFL(INDEXJ)                                         SIFT   18
      XCVLE(INDEXI) = XCVLE(INDEXJ)                                     SIFT   19
      XCVTE(INDEXI) = XCVTE(INDEXJ)                                     SIFT   20
      INDEXI = INDEXJ                                                   SIFT   21
      INDEXJ = INDEXI * 2                                               SIFT   22
      GO TO 10                                                          SIFT   23
C                                                                       SIFT   24
   30 YFL(INDEXI) = Y1                                                  SIFT   25
      XCVLE(INDEXI) = X1                                                SIFT   26
      XCVTE(INDEXI) = X2                                                SIFT   27
      RETURN                                                            SIFT   28
      END                                                               SIFT   29
      SUBROUTINE FTLUP (X,Y,M,N,VARI,VARD)                              FTLUP1 2
C     ***DOCUMENT DATE 09-12-69    SUBROUTINE REVISED 07-07-69 *********FTLUP1 3
C        MODIFICATION OF LIBRARY INTERPOLATION SUBROUTINE  FTLUP        FTLUP1 4
      DIMENSION VARI(1), VARD(1), V(3), YY(2)                           FTLUP1 5
      DIMENSION II(43)                                                  FTLUP1 6
C                                                                       FTLUP1 7
C      INITIALIZE ALL INTERVAL POINTERS TO -1.0   FOR MONOTONICITY CHECKFTLUP1 8
      DATA (II(J),J=1,43)/43*-1/                                        FTLUP1 9
      MA=IABS(M)                                                        FTLUP110
C                                                                       FTLUP111
C           ASSIGN INTERVAL POINTER FOR GIVEN VARI TABLE                FTLUP112
C      THE SAME POINTER WILL BE USED ON A GIVEN VARI TABLE EVERY TIME   FTLUP113
      LI=MOD(LOCF(VARI(1)),43)+1                                        FTLUP114
      I=II(LI)                                                          FTLUP115
      IF (I.GE.0) GO TO 60                                              FTLUP116
      IF (N.LT.2) GO TO 60                                              FTLUP117
C                                                                       FTLUP118
C     MONOTONICITY CHECK                                                FTLUP119
      IF (VARI(2)-VARI(1)) 20,20,40                                     FTLUP120
C     ERROR IN MONOTONICITY                                             FTLUP121
10    K=LOCF(VARI(1))                                                   FTLUP122
      PRINT 170, J,K,(VARI(J),J=1,N),(VARD(J),J=1,N)                    FTLUP123
      STOP                                                              FTLUP124
C     MONOTONIC DECREASING                                              FTLUP125
20    DO 30 J=2,N                                                       FTLUP126
      IF (VARI(J)-VARI(J-1)) 30,10,10                                   FTLUP127
30    CONTINUE                                                          FTLUP128
```

```
      GO TO 60                                                          FTLUP129
C     MONOTONIC INCREASING                                              FTLUP130
40    DO 50 J=2,N                                                       FTLUP131
      IF (VARI(J)-VARI(J-1)) 10,10,50                                   FTLUP132
50    CONTINUE                                                          FTLUP133
C                                                                       FTLUP134
C     INTERPOLATION                                                     FTLUP135
60    IF (I.LE.0) I=1                                                   FTLUP136
      IF (I.GE.N) I=N-1                                                 FTLUP137
      IF (N.LE.1) GO TO 70                                              FTLUP138
      IF (MA.NE.0) GO TO 80                                             FTLUP139
C     ZERO ORDER                                                        FTLUP140
70    Y=VARD(1)                                                         FTLUP141
      GO TO 160                                                         FTLUP142
C     LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))                           FTLUP143
80    IF ((VARI(I)-X)*(VARI(I+1)-X)) 110,110,90                         FTLUP144
C     IN GIVES DIRECTION FOR SEARCH OF INTERVALS                        FTLUP145
90    IN=SIGN(1.0,(VARI(I+1)-VARI(I))*(X-VARI(I)))                      FTLUP146
C     IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL             FTLUP147
100   IF ((I+IN).LE.0) GO TO 110                                        FTLUP148
      IF ((I+IN).GE.N) GO TO 110                                        FTLUP149
      I=I+IN                                                            FTLUP150
      IF ((VARI(I)-X)*(VARI(I+1)-X)) 110,110,100                        FTLUP151
110   IF (MA.EQ.2) GO TO 120                                            FTLUP152
C                                                                       FTLUP153
C     FIRST ORDER                                                       FTLUP154
      Y=(VARD(I)*(VARI(I+1)-X)-VARD(I+1)*(VARI(I)-X))/(VARI(I+1)-VARI(I)FTLUP155
     1)                                                                 FTLUP156
      GO TO 160                                                         FTLUP157
C                                                                       FTLUP158
C     SECOND ORDER                                                      FTLUP159
120   IF (N.EQ.2) GO TO 10                                              FTLUP160
      IF (I.EQ.(N-1)) GO TO 140                                         FTLUP161
      IF (I.EQ.1) GO TO 130                                             FTLUP162
C     PICK THIRD POINT                                                  FTLUP163
      SK=VARI(I+1)-VARI(I)                                              FTLUP164
      IF ((SK*(X-VARI(I-1))).LT.(SK*(VARI(I+2)-X))) GO TO 140           FTLUP165
130   L=I                                                               FTLUP166
      GO TO 150                                                         FTLUP167
140   L=I-1                                                             FTLUP168
150   V(1)=VARI(L)-X                                                    FTLUP169
      V(2)=VARI(L+1)-X                                                  FTLUP170
      V(3)=VARI(L+2)-X                                                  FTLUP171
      YY(1)=(VARD(L)*V(2)-VARD(L+1)*V(1))/(VARI(L+1)-VARI(L))           FTLUP172
      YY(2)=(VARD(L+1)*V(3)-VARD(L+2)*V(2))/(VARI(L+2)-VARI(L+1))       FTLUP173
      Y=(YY(1)*V(3)-YY(2)*V(1))/(VARI(L+2)-VARI(L))                     FTLUP174
160   II(LI)=I                                                          FTLUP175
      RETURN                                                            FTLUP176
C                                                                       FTLUP177
```

```
C                                                                       FTLUP178
170   FORMAT (1H1,50H TABLE BELOW OUT OF ORDER FOR FTLUP  AT POSITION  ,FTLUP179
     1I5,/31H X TABLE IS STORED IN LOCATION ,O6,//(8G15.8))             FTLUP180
      END                                                               FTLUP181
      SUBROUTINE CDICLS (AR,ARTRUE,ISEMSP,MTOT,NSV,CDI,CDIT)            CDICLS 2
      DIMENSION ETAN(51), GAMPR(51,1), ETA(41), GAMMA(41), VE(41), B(41)CDICLS 3
     1, FVN(41,41)                                                      CDICLS 4
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                ALL     2
     $    STA(4), TBLSCW(100), YYCP(4),                                 ALL     3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),         ALL     4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                        ALL     5
C                                                                       ALL     6
      COMMON /THRECDI/ SLOAD(3,100)                                     CDICLS 6
      DO 10 I=1,41                                                      CDICLS 7
      DO 10 J=1,41                                                      CDICLS 8
10    FVN(I,J)=0                                                        CDICLS 9
      SPAN=2.*BOT                                                       CDICLS10
      CAVB=SPAN/ARTRUE                                                  CDICLS11
      PI=.314159265E+01                                                 CDICLS12
      NST=ISEMSP+1                                                      CDICLS13
      NN=MTOT                                                           CDICLS14
      DO 20 N=1,ISEMSP                                                  CDICLS15
      NM=NSV-N                                                          CDICLS16
      NSCW=TBLSCW(NM)                                                   CDICLS17
      NN=NN-NSCW                                                        CDICLS18
      ETAN(N)=ASIN(-Q(NN)*2./SPAN)                                      CDICLS19
      GAMPR(N,1)=SLOAD(3,NM)*CAVB/(2.*SPAN)                             CDICLS20
20    CONTINUE                                                          CDICLS21
      ETAN(NST)=PI/2.                                                   CDICLS22
      GAMPR(NST,1)=0                                                    CDICLS23
      DO 30 NP=1,41                                                     CDICLS24
      ANP=NP                                                            CDICLS25
30    ETA(NP)=(ANP-21.)*PI/42.                                          CDICLS26
C                                                                       CDICLS27
      DO 40 JK=21,41                                                    CDICLS28
      CALL FTLUP (ETA(JK),GAMMA(JK),1,NST,ETAN,GAMPR)                   CDICLS29
40    CONTINUE                                                          CDICLS30
      DO 50 NY=22,41                                                    CDICLS31
      ETA(NY)=SIN(ETA(NY))                                              CDICLS32
      NR=42-NY                                                          CDICLS33
      ETA(NR)=-ETA(NY)                                                  CDICLS34
50    GAMMA(NR)=GAMMA(NY)                                               CDICLS35
      DO 90 NU=21,41                                                    CDICLS36
      ANU=NU                                                            CDICLS37
      DO 80 N=1,41                                                      CDICLS38
      AN=N                                                              CDICLS39
      NNUD=IABS(N-NU)                                                   CDICLS40
      VE(N)=COS(((AN-21.)*PI)/42.)                                      CDICLS41
      IF (NNUD.NE.0) GO TO 60                                           CDICLS42
```

```
      B(N)=(42.)/(4.0*COS(((ANU-21.)*PI)/42.))                   CDICLS43
      GO TO 80                                                   CDICLS44
 60   IF (MOD(NNUD,2).EQ.0) GO TO 70                             CDICLS45
      B(N)=VE(N)/((42.)*(ETA(N)-ETA(NU))**2)                     CDICLS46
      GO TO 80                                                   CDICLS47
 70   B(N)=0.0                                                   CDICLS48
 80   CONTINUE                                                   CDICLS49
      DO 90 NP=21,41                                             CDICLS50
      NUST=IABS(NU-21)                                           CDICLS51
      IF (NUST.EQ.0) GO TO 90                                    CDICLS52
      IF (MOD(NUST,2).EQ.0) GO TO 90                             CDICLS53
      NPST=IABS(NP-20)                                           CDICLS54
      IF (MOD(NPST,2).EQ.0) GO TO 90                             CDICLS55
      NPNUD=IABS(NP-NU)                                          CDICLS56
      IF (NPNUD.EQ.0) GO TO 90                                   CDICLS57
      IF (MOD(NPNUD,2).EQ.0) GO TO 90                            CDICLS58
      FVN(NU,NP)=2.0*B(NP)/21.*COS((ANU-21.)*PI/42.)             CDICLS59
      IT=42-NU                                                   CDICLS60
      ITT=42-NP                                                  CDICLS61
      FVN(NU,ITT)=2.0*B(ITT)/21.*COS((ANU-21.)*PI/42.)           CDICLS62
      FVN(IT,NP)=FVN(NU,ITT)                                     CDICLS63
      FVN(IT,ITT)=FVN(NU,NP)                                     CDICLS64
 90   CONTINUE                                                   CDICLS65
C                                                                CDICLS66
      CCC=0.0                                                    CDICLS67
      DO 100 N=1,41                                              CDICLS68
 100  CCC=CCC+(GAMMA(N)*GAMMA(N))                                CDICLS69
      CCD=0.0                                                    CDICLS70
      DO 110 NUP=1,41                                            CDICLS71
      DO 110 N=1,41                                              CDICLS72
      CCD=CCD-2.0*FVN(NUP,N)*(GAMMA(NUP)*GAMMA(N))               CDICLS73
 110  CONTINUE                                                   CDICLS74
      CDI=PI*AR/4.*(CCC+CCD)                                     CDICLS75
      CDIT=1./(PI*AR)                                            CDICLS76
      RETURN                                                     CDICLS77
      END                                                        CDICLS78
      SUBROUTINE CDRAGNF                                         CDRAGNF2
      DIMENSION GAM(1100),XC4(1100),YQ(1100),CCR(40),            CDRAGNF3
     $ FW(2),FV(2),XXCC(40),CCC(400),CRR(400),                   CDRAGNF4
     $ YB(100),CRI(102),NMA(4),XCC4(400),CHD(100),               CDRAGNF5
     $ XC44(100),YY(2),PPHI(100),ZZH(100),Z(1100),               CDRAGNF6
     $ PHII(1000),SA(100),SSA(1100),ALUP(400),ALLP(100),         CDRAGNF7
     $ ALPPD(1100),ALD(40),YC(102),YQQ(100),BOTL(4),NSUMSV(4),   CDRAGNF8
     $ MSUMSV(4),NMSUMS(4),NUMBER(4)                             CDRAGNF9
      COMMON /ALL/ POT, BOTSV(4), M, BETA, PTEST, QTEST,         ALL     2
     $    STA(4), TBLSCW(100), YYCP(4),                          ALL     3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),  ALL     4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                 ALL     5
C                                                                ALL     6
```

```
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,      ONETHRE2
     $   ARTRUE, RTCDHT(4), CONFIG(2), NSSWSV(4),                         ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                                 ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),        ONETHRE5
     $ CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                             ONETHRE6
     $ YINNER(4), YOUTER(4)                                               ONETHRE7
C                                                                         ONETHRE8
      INTEGER CONFIG                                                      ONETHRE9
C                                                                         ONETHR10
      COMMON /TOTHREE/ CIR(400,2)                                         TOTHRE 2
C                                                                         TOTHRE 3
      COMMON /INSUB23/ APSI,APHI,XX,YYY,ZZ,SNN,TOLCSQ                     CDRAGN13
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD                 THREFOR2
C                                                                         THREFOR3
      COMMON/CCRRDD/ TSPAN(4), TSPANA, KBIT, CTILDA, XTILDA, DISTALE      CCRRDD 2
C                                                                         CCRRDD 3
C                                                                         CDRAGN16
      DATA NUMBER/5HFIRST,6HSECOND,5HTHIRD,6HFOURTH/                      CDRAGN17
C                                                                         CDRAGN18
      WRITE (6,250)                                                       CDRAGN19
      APSI=TOLCSQ=TBLS=0.                                                 CDRAGN20
      PI=4.*ATAN(1.)                                                      CDRAGN21
      FPI=4.*PI                                                           CDRAGN22
      DO 5 IT=1,IPLAN                                                     CDRAGN23
      BOTL(IT) = ABS(TSPAN(IT))                                           CDRAGN24
    5 CONTINUE                                                            CDRAGN25
      SNN = BOTL(KBOT) / ( 2. * NSSWSV(KBOT))                             CDRAGN26
      DELTYB=2.*SNN                                                       CDRAGN27
      NSMS = 0                                                            CDRAGN28
      MSMS = 0                                                            CDRAGN29
      NMSMS = 0                                                           CDRAGN30
      NMAX = 0                                                            CDRAGN31
      DO 7 IT=1,IPLAN                                                     CDRAGN32
      NMA(IT) = BOTL(IT) / DELTYB                                         CDRAGN33
      NSMS = NSMS + NSSWSV(IT)                                            CDRAGN34
      NSUMSV(IT) = NSMS                                                   CDRAGN35
      MSMS = MSMS + MSV(IT)                                               CDRAGN36
      MSUMSV(IT) = MSMS                                                   CDRAGN37
      NMSMS = NMSMS + NMA(IT)                                             CDRAGN38
      NMSUMS(IT) = NMSMS                                                  CDRAGN39
    7 CONTINUE                                                            CDRAGN40
      NMAX = NMSUMS(IPLAN)                                                CDRAGN41
      DO 10 I=1,M                                                         CDRAGN42
      CRR(I)=CIR(I,1)+CIR(I,2)*(CLDES-CLT)/CLNT                           CDRAGN43
   10 CONTINUE                                                            CDRAGN44
      SCWMIN=20.                                                          CDRAGN45
      DO 20 I=1,NSSW                                                      CDRAGN46
   20 SCWMIN=AMIN1(SCWMIN,TBLSCW(I))                                      CDRAGN47
      NSCWMIN=SCWMIN                                                      CDRAGN48
```

```
      MM=NSCWMIN*NMAX                                                  CDRAGN49
      DELTXOC=1./SCWMIN                                                CDRAGN50
      DO 100 LA=1,NSSW                                                 CDRAGN51
      CHD(LA)=CCAV(2,LA)*CAVE/BETA                                     CDRAGN52
      DELTXX=1./TBLSCW(LA)                                             CDRAGN53
      XC=-.75*DELTXX                                                   CDRAGN54
      ITBL=TBLSCW(LA)                                                  CDRAGN55
      DO 30 LB=1,ITBL                                                  CDRAGN56
      XC=XC+DELTXX                                                     CDRAGN57
      XXCC(LB)=XC                                                      CDRAGN58
      LC=LB+TBLS                                                       CDRAGN59
   30 ALO(LB)=ALP(LC)                                                  CDRAGN60
      XLE=PN(LC)+CHD(LA)*(1.-.75/TBLSCW(LA))                           CDRAGN61
      XOC=-.75*DELTXOC                                                 CDRAGN62
      KCODE=LB=0                                                       CDRAGN63
      DO 90 K=1,NSCWMIN                                                CDRAGN64
      J=K+(LA-1)*NSCWMIN                                               CDRAGN65
      XOC=XOC+DELTXOC                                                  CDRAGN66
      XCC4(J)=-XOC*CHD(LA)+XLE                                         CDRAGN67
      CALL FILUP (XOC,ALOP(J),+1,ITBL,XXCC,ALO)                        CDRAGN68
      AXMN=K*DELTXOC                                                   CDRAGN69
      CAT=0.                                                           CDRAGN70
      IF (KCODE.EQ.2) CAT=CCR(LB)-CUT                                  CDRAGN71
      KCODE=0                                                          CDRAGN72
   40 LB=LB+1                                                          CDRAGN73
      LC=LB+TBLS                                                       CDRAGN74
      CCR(LB)=CRR(LC)                                                  CDRAGN75
      AXITBL=LB*DELTXX                                                 CDRAGN76
      IF (AXMN-AXITBL) 50,60,70                                        CDRAGN77
   50 CUT=CCR(LB)*(AXMN-(LB-1)*DELTXX)/DELTXX                          CDRAGN78
      KCODE=2                                                          CDRAGN79
      GO TO 80                                                         CDRAGN80
   60 KCODE=1                                                          CDRAGN81
   70 CUT=CCR(LB)                                                      CDRAGN82
   80 CAT=CAT+CUT                                                      CDRAGN83
      IF (KCODE.GE.1) GO TO 90                                         CDRAGN84
      IF (LB.LT.ITBL) GO TO 40                                         CDRAGN85
   90 CCC(J)=CAT                                                       CDRAGN86
      TBLS=TBLS+TBLSCW(LA)                                             CDRAGN87
  100 CONTINUE                                                         CDRAGN88
      II=1                                                             CDRAGN89
      DO 150 I=1,IPLAN                                                 CDRAGN90
      IUZ=NSSWSV(I)                                                    CDRAGN91
      IUX=IUZ+1                                                        CDRAGN92
      IF (STLCIND(I) .EQ. 1.) IUX = IUZ                                CDRAGN93
      IC = NSUMSV(I)                                                   CDRAGN94
      ID=IC+1                                                          CDRAGN95
      IZ = NSUMSV(I)                                                   CDRAGN96
      YCAT=0.                                                          CDRAGN97
```

```
      IAMM=NMA(I)                                                   CDRAGN98
      IF(I.EQ.1) GOTO 142                                           CDRAGN99
      NST = NSUMSV(I-1)                                             CDRAG100
      NSTT = NMSUMS(I-1)                                            CDRAG101
      GOTO 143                                                      CDRAG102
  142 NST = 0                                                       CDRAG103
      NSTT = 0                                                      CDRAG104
  143 CONTINUE                                                      CDRAG105
      DO 140 LA=1,NSCWMIN                                           CDRAG106
      IF (STLOIND(I) .EQ. 1.) GO TO 144                             CDRAG107
      YC(1)=-PI/2.                                                  CDRAG108
      CRI(1)=0.                                                     CDRAG109
  144 DO 120 J= 1,IUZ                                               CDRAG110
      L=J+1                                                         CDRAG111
      IF (STLOIND(I) .EQ. 1.) L = J                                 CDRAG112
      LU = LA + (J-1 + NST) * NSCWMIN                               CDRAG113
      ALLP(J)=ALOP(LU)                                              CDRAG114
      XC44(J)=XCC4(LU)                                              CDRAG115
      CRI(L)=CCC(LU)                                                CDRAG116
      IF (LA.NE.1) GO TO 120                                        CDRAG117
      JJ = J + NST                                                  CDRAG118
      ZZH(J)=ZH(JJ)                                                 CDRAG119
      SA(J)=SSWWA(JJ)                                               CDRAG120
      PPHI(J)=PHI(JJ)                                               CDRAG121
      YQQ(J)=Q(II)                                                  CDRAG122
      II=II+TBLSCW(JJ)                                              CDRAG123
      IE=IUZ-J+1                                                    CDRAG124
      ITL=TBLSCW(IZ)                                                CDRAG125
      ID=ID-ITL                                                     CDRAG126
      IA=ID+ITL                                                     CDRAG127
      IF (IA.GT.IC) YCAT=YCAT-S(ID)                                 CDRAG128
      IF (IA.GT.IC) GO TO 110                                       CDRAG129
      YCAT=YCAT-S(ID)-S(IA)                                         CDRAG130
110   IZ=IZ-1                                                       CDRAG131
      YB(IE)=YCAT                                                   CDRAG132
120   CONTINUE                                                      CDRAG133
      DO 130 JP=1,IUZ                                               CDRAG134
      JZ=JP+1                                                       CDRAG135
      IF (STLOIND(I) .EQ. 1.) JZ = JP                               CDRAG136
      YC(JZ) = ASIN(YB(JP)/BOTL(I))                                 CDRAG137
130   CONTINUE                                                      CDRAG138
      YOB=-NMA(I)*2.*SNN-SNN                                        CDRAG139
      DO 140 K=1,IAMM                                               CDRAG140
      KP = LA + (K-1 + NSTT) * NSCWMIN                              CDRAG141
      YOB=YOB+DELTYB                                                CDRAG142
      YOC = ASIN(YOB/BOTL(I))                                       CDRAG143
      CALL FTLUP (YOB,YQ(KP),+1,IUZ,YB,YQQ)                         CDRAG144
      CALL FTLUP (YOB,ALPPD(KP),+1,IUZ,YB,ALLP)                     CDRAG145
      CALL FTLUP (YOB,SSA(KP),+1,IUZ,YB,SA)                         CDRAG146
```

```
      CALL FTLUP (YOB,XC4(KP),+1,IUZ,YB,XC44)                         CDRAG147
      CALL FTLUP (YOB,Z(KP),+1,IUZ,YB,ZZH)                            CDRAG148
      CALL FTLUP (YOB,PHII(KP),+1,IUZ,YB,PPHI)                        CDRAG149
      CALL FTLUP (YOC,GAM(KP),+1,IUX,YC,CRI)                          CDRAG150
      IF (YOB.GT.YB(IUZ)) GAM(KP)=CRI(IUX)                            CDRAG151
140   CONTINUE                                                        CDRAG152
150   CONTINUE                                                        CDRAG153
      CDRAG=CTHRUST=CSUCT=0.                                          CDRAG154
      CONST=16.*SNN*BOT/SREF                                          CDRAG155
      DO 190 LI=1,NMAX                                                CDRAG156
      LA=(LI-1)*NSCWMIN+1                                             CDRAG157
      LB=LI*NSCWMIN                                                   CDRAG158
      CDRAGIT=CTT=0.                                                  CDRAG159
      DO 180 NV=LA,LB                                                 CDRAG160
      CPT=COS(ATAN(PHII(NV)))                                         CDRAG161
      VELIN=0.                                                        CDRAG162
      DO 170 NN=1,MM                                                  CDRAG163
      XX=XC4(NV)-XC4(NN)                                              CDRAG164
      YY(1)=YQ(NV)-YQ(NN)                                             CDRAG165
      YY(2)=YQ(NV)+YQ(NN)                                             CDRAG166
      ZZ=Z(NV)-Z(NN)                                                  CDRAG167
      APHI=ATAN(PHII(NN))                                             CDRAG168
      DO 160 I=1,2                                                    CDRAG169
      YYY=YY(I)                                                       CDRAG170
      CALL INFSUB (BOT,FV(I),FW(I),FUI)                               CDRAG171
      APHI=-APHI                                                      CDRAG172
160   CONTINUE                                                        CDRAG173
      VELIN=((FW(1)+FW(2))-(FV(1)+FV(2))*PHII(NV))*GAM(NN)/FPI+VELIN  CDRAG174
170   CONTINUE                                                        CDRAG175
      CTT=CTT+GAM(NV)*(ALPD/57.29578+ALPPD(NV))*CPT/(2.*BOT)          CDRAG176
180   CDRAGIT=CDRAGIT+VELIN*GAM(NV)*CPT/(2.*BOT)                      CDRAG177
      CTT=CTT-CDRAGIT                                                 CDRAG178
      SWLE=ATAN(SSA(LA))                                              CDRAG179
      CST=CTT/COS(SWLE)                                               CDRAG180
      CCC(LI)=CDRAGIT                                                 CDRAG181
      CRR(LI)=CTT                                                     CDRAG182
      XCC4(LI)=CST                                                    CDRAG183
      CDRAG=CDRAG+CDRAGIT*CONST                                       CDRAG184
      CTHRUST=CTHRUST+CTT*CONST                                       CDRAG185
      CSUCT=CSUCT+CST*CONST                                           CDRAG186
190   CONTINUE                                                        CDRAG187
      TBLE=II=0                                                       CDRAG188
      LI=0                                                            CDRAG189
      LBLR=0                                                          CDRAG190
      DO 220 I=1,IPLAN                                                CDRAG191
      IAMM=NMA(I)                                                     CDRAG192
      IF(I.EQ.1) GOTO 225                                             CDRAG193
      NST = NSUMSV(I-1)                                               CDRAG194
      NSTT = NMSUMS(I-1)                                              CDRAG195
```

```
      GOTO 226                                                          CDRAG196
  225 NST = 0                                                           CDRAG197
      NSTT = 0                                                          CDRAG198
  226 CONTINUE                                                          CDRAG199
      DO 200 J=1,IAMM                                                   CDRAG200
      JJ = J + NSTT                                                     CDRAG201
      LA = 1 + (J-1 + NSTT) * NSCWMIN                                   CDRAG202
      GAM(J)=CCC(JJ)                                                    CDRAG203
      XC4(J)=CRR(JJ)                                                    CDRAG204
      Z(J)=XCC4(JJ)                                                     CDRAG205
      PHII(J)=YQ(LA)                                                    CDRAG206
  200 CONTINUE                                                          CDRAG207
      IUZ=NSSWSV(I)                                                     CDRAG208
      DO 210 LBLAIR=1,IUZ                                               CDRAG209
      LI=LI+1                                                           CDRAG210
      LU=1+TBLE                                                         CDRAG211
      LBLR=LBLR+1                                                       CDRAG212
      YBB=Q(LU)                                                         CDRAG213
      II=II+1                                                           CDRAG214
      TBLE=TBLE+TBLSCW(II)                                              CDRAG215
      YOOB=YBB/BOT                                                      CDRAG216
      CALL FTLUP (YBB,CDRAGIT,+1,IAMM,PHII,GAM)                         CDRAG217
      CALL FTLUP (YBB,CTT,+1,IAMM,PHII,XC4)                             CDRAG218
      CALL FTLUP (YBB,CST,+1,IAMM,PHII,Z)                               CDRAG219
      LL = LBLAIR + NST                                                 CDRAG220
      SWALE=ATAN(SSWWA(LL))*57.29578                                    CDRAG221
      DO 300 IT = 1,IPLAN                                               CDRAG222
      IF(IT.GT.1) GOTO 310                                              CDRAG223
      MOD = 1                                                           CDRAG224
      GOTO 320                                                          CDRAG225
  310 MOD = NSUMSV(IT-1) + 1                                            CDRAG226
  320 IF(II.EQ.MOD) WRITE(6,240) NUMBER(IT)                             CDRAG227
  300 CONTINUE                                                          CDRAG228
      WRITE (6,260) LI,YOOB,SWALE,CDRAGIT,CTT,CST                       CDRAG229
      BLAIR(LBLR)=CST                                                   CDRAG230
  210 CONTINUE                                                          CDRAG231
  220 CONTINUE                                                          CDRAG232
  230 CDCL2=CDRAG/CLDES**2                                              CDRAG233
      WRITE (6,270) CDCL2,CTHRUST,CSUCT                                 CDRAG234
C                                                                       CDRAG235
C                                                                       CDRAG236
C                                                                       CDRAG237
  240 FORMAT(/37X,20HCONTRIBUTION OF THE ,A10,                          CDRAG238
     *35HPLANFORM TO THE CHORD OR DRAG FORCE/)                          CDRAG239
  250 FORMAT (////30X,73HINDUCED DRAG, LEADING EDGE THRUST AND SUCTION CCDRAG240
     1OEFFICIENT CHARACTERISTICS/40X,53HCOMPUTED AT THE DESIRED CL FROM CDRAG241
     2A NEAR FIELD SOLUTION//58X,20HSECTION COEFFICIENTS/48X,11HL. E. SWCDRAG242
     3EEP/25X,7HSTATION,9X,5H 2Y/B,5X,5HANGLE,7X,9HCDII C/2B,5X,7HCT C/2CDRAG243
     4B,5X,7HCS C/2B)                                                   CDRAG244
```

```
260   FORMAT (20X,I10,5X,5F12.5)                                        CDRAG245
270   FORMAT (///57X,18HTOTAL COEFFICIENTS//36X,12HCDII/CL**2 =F10.5,5X,CDRAG246
     13HCT=,F10.5,5X,3HCS=,F10.5)                                        CDRAG247
      END                                                                CDRAG248
      SUBROUTINE TIPSUCT                                                 TIPSUCT2
      DIMENSION YY(2), WVOU(60), FV(2), FW(2), XTLEG(60),                TIPSUCT3
     $          ZLEGSV(50,4), CIRSUM(50,4), YLEGSV(50,4)                 TIPSUCT4
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                 ALL     2
     $    STA(4), TBLSCW(100), YYCP(4),                                  ALL     3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),          ALL     4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                         ALL     5
C                                                                        ALL     6
C                                                                        TIPSUCT6
C                                                                        NOTES   2
C          ---  NOTES TO THE USERS  ---                                  NOTES   3
C                                                                        NOTES   4
C                                                                        NOTES   5
C      1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING                  NOTES   6
C         EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF                 NOTES   7
C         ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE                    NOTES   8
C         CALCULATED FROM CLDES = 100. AND CLDES = 1.0                   NOTES   9
C                                                                        NOTES  10
C                                                                        NOTES  11
C       2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOM-      NOTES  12
C          MENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE     NOTES  13
C         MORE MEANINGFUL SIDE EDGE RESULTS                              NOTES  14
C                                                                        NOTES  15
C                                                                        NOTES  16
C                                                                        NOTES  17
C       3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO           NOTES  18
C          0. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE           NOTES  19
C          EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO         NOTES  20
C          1.0 IF THIS LOADING IS TO BE NON-ZERO                         NOTES  21
C                                                                        NOTES  22
C                                                                        NOTES  23
C                                                                        TIPSUCT8
      COMMON /TOTHREE/ CIR(400,2)                                        TOTHRE  2
C                                                                        TOTHRE  3
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,     ONETHRE2
     $   ARTRUE, RTCDHT(4), CONFIG(2), NSSWSV(4),                        ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                                ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),       ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                           ONETHRE6
     $  YINNER(4), YOUTER(4)                                             ONETHRE7
C                                                                        ONETHRE8
      INTEGER CONFIG                                                     ONETHRE9
C                                                                        ONETHR10
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD                THREFOR2
C                                                                        THREFOR3
```

```
      COMMON /INSUB23/ APSI,APHI,XX,YYY,ZZ,SNN,TOLCSQ                    TIPSUC12
      DIMENSION CLLL(26),CDDD(26),CMMM(26),ALPH(26),CDCLZ(26),CDCLF(26),TIPSUC13
     1          CDCLV(26)                                                 TIPSUC14
      REAL KVSE(4)                                                        TIPSUC15
      DIMENSION CENTR(4), TIPSUM(4)                                       TIPSUC16
      DATA XTLEG / 60*0.0/                                                TIPSUC17
      DATA ZLEGSV, CIRSUM, YLEGSV                                         TIPSUC18
     $  / 200*0.0, 200*0.0, 200*0.0 /                                    TIPSUC19
      J = 0                                                               TIPSUC20
      DO 1 ITT = 1,IPLAN                                                  TIPSUC21
        CENTR(ITT) = 0.0                                                  TIPSUC22
        KVSE(ITT) = 0.0                                                   TIPSUC23
        TIPSUM(ITT) = 0.0                                                 TIPSUC24
        IF(XL(ITT) .EQ. XT(ITT)) J = J + 1                                TIPSUC25
    1 CONTINUE                                                            TIPSUC26
C                                                                         TIPSUC27
      IF(J .EQ. IPLAN)GO TO 540                                           TIPSUC28
C                                                                         TIPSUC29
      BLAMAR = 1.0/BETA                                                   TIPSUC30
      NSSW = 0                                                            TIPSUC31
      DO 2 ITT = 1,IPLAN                                                  TIPSUC32
        XT(ITT) = XT(ITT) * BLAMAR                                        TIPSUC33
        XL(ITT) = XL(ITT) * BLAMAR                                        TIPSUC34
        NSSW = NSSW + NSSWSV(ITT)                                         TIPSUC35
    2 CONTINUE                                                            TIPSUC36
C                                                                         TIPSUC37
C                                                                         TIPSUC38
C     THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE       TIPSUC39
C     CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES                TIPSUC40
C                                                                         TIPSUC41
      TOLC=.0100*BOT                                                      TIPSUC42
      TOLCSQ=TOLC*TOLC                                                    TIPSUC43
C                                                                         TIPSUC44
      TIPSI = 0.0                                                         TIPSUC45
C                                                                         TIPSUC46
      TIPSU=PITCH=0.                                                      TIPSUC47
C                                                                         TIPSUC48
C                                                                         TIPSUC49
C     GEOMETRY FOR TIP TRAILING LEGS                                      TIPSUC50
C                                                                         TIPSUC51
      IM = 0                                                              TIPSUC52
      IMM = 0                                                             TIPSUC53
      NSSW1 = 0                                                           TIPSUC54
      NSSW2 = 0                                                           TIPSUC55
      EPS = 1.E-6                                                         TIPSUC56
      CTSW = 0.0                                                          TIPSUC57
      CMW = 0.0                                                           TIPSUC58
      CCIRS = 0.0                                                         TIPSUC59
C                                                                         TIPSUC60
```

```
      DO 535 ITT = 1,IPLAN                                              TIPSUC61
C                                                                       TIPSUC62
      NSCW = MSV(ITT)/NSSWSV(ITT)                                       TIPSUC63
      NSSW2 = NSSW2 + NSSWSV(ITT)                                       TIPSUC64
      IF(ITT .EQ. IPLAN .AND. XL(ITT) .EQ. XT(ITT)) GO TO 475           TIPSUC65
20    I=IMM+1                                                           TIPSUC66
      IMP = 0                                                           TIPSUC67
      J=IMM+2                                                           TIPSUC68
      IUU = 2                                                           TIPSUC69
      APHI=ATAN(PHI(IM+1))                                              TIPSUC70
   25 SA = SIN(APHI)                                                    TIPSUC71
      CA=COS(APHI)                                                      TIPSUC72
      TLX1=PN(I)-S(I)*TAN(PSI(I))*CA                                    TIPSUC73
      TLX2=PN(J)-S(J)*TAN(PSI(J))*CA                                    TIPSUC74
      CLFTLG=TLX1-TLX2                                                  TIPSUC75
      XTLEG(1)=TLX1/2.+TLX2/2.                                          TIPSUC76
      YLEG=Q(I)-S(I)*CA                                                 TIPSUC77
      YLEGSV(1,ITT) = YLEG                                              TIPSUC78
      ZLEG=ZH(IM+1)-S(I)*SA                                             TIPSUC79
      ZLEGSV(1,ITT) = ZLEG                                              TIPSUC80
      IF (XL(ITT).EQ.XT(ITT)) GO TO 100                                 TIPSUC81
      DO 30 NV=2,NSCW                                                   TIPSUC82
      NVT=NV-1                                                          TIPSUC83
30    XTLEG(NV)=XTLEG(NVT)-CLFTLG                                       TIPSUC84
      NCTL=0                                                            TIPSUC85
      NA=1                                                              TIPSUC86
      NB=NSCW                                                           TIPSUC87
40    DO 70 NV=NA,NB                                                    TIPSUC88
C                                                                       TIPSUC89
C                                                                       TIPSUC90
C     THE RATIO OF W/U IS INITIALIZED TO -1 BECAUSE IN THE TERM         TIPSUC91
C     -U*ALPHA/U,USED IN THIS SUMMATION, ALPHA IS SET TO 1 RADIAN       TIPSUC92
C     SO THAT THE RESULTING TIP SUCTION CAN BE USED DIRECTLY TO FIND    TIPSUC93
C     KV SIDE EDGE                                                      TIPSUC94
C                                                                       TIPSUC95
C                                                                       TIPSUC96
      WVOU(NV)=-1.                                                      TIPSUC97
      IZ=1                                                              TIPSUC98
      NNN=TBLSCW(IZ)                                                    TIPSUC99
      DO 60 NN=1,M                                                      TIPSU100
      APHI=ATAN(PHI(IZ))                                                TIPSU101
      APSI=PSI(NN)                                                      TIPSU102
      XX=XTLEG(NV)-PN(NN)                                               TIPSU103
      YY(1)=YLEG-Q(NN)                                                  TIPSU104
      YY(2)=YLEG+Q(NN)                                                  TIPSU105
      ZZ=ZLEG-ZH(IZ)                                                    TIPSU106
      SNN=S(NN)                                                         TIPSU107
      DO 50 I=1,2                                                       TIPSU108
      YYY=YY(I)                                                         TIPSU109
```

```
      CALL INFSUB (BOT,FV(I),FW(I),FUI)                                 TIPSU110
      APHI = -APHI                                                      TIPSU111
      APSI = -APSI                                                      TIPSU112
50    CONTINUE                                                          TIPSU113
      WVOU(NV)=WVOU(NV)+(FW(1)+FW(2))*CIR(NN,2)/12.5663704              TIPSU114
      IF (NN.LT.NNN.OR.NN.EQ.M) GO TO 60                                TIPSU115
      IZ=IZ+1                                                           TIPSU116
      NNN=NNN+TBLSCW(IZ)                                                TIPSU117
60    CONTINUE                                                          TIPSU118
70    CONTINUE                                                          TIPSU119
      NCTL=NCTL+1                                                       TIPSU120
      IF (NCTL-2) 80,100,150                                            TIPSU121
C                                                                       TIPSU122
C     GEOMETRY FOR SPANWISE BOUND VORTICES                              TIPSU123
C                                                                       TIPSU124
80    NA=NSCW+1                                                         TIPSU125
      NB=2*NSCW                                                         TIPSU126
      JA=IMM+1                                                          TIPSU127
      YLEG=Q(JA)                                                        TIPSU128
      ZLEG=ZH(IM+1)                                                     TIPSU129
      DO 90 J=1,NSCW                                                    TIPSU130
      JK=IMM+J                                                          TIPSU131
      NV=J+NSCW                                                         TIPSU132
90    XTLEG(NV)=PN(JK)                                                  TIPSU133
      GO TO 40                                                          TIPSU134
C                                                                       TIPSU135
C     GEOMETRY ALONG RIGHT TRAILING LEGS                                TIPSU136
C                                                                       TIPSU137
100   NA=2*NSCW+1                                                       TIPSU138
      NB=3*NSCW                                                         TIPSU139
      CCIRS=0.                                                          TIPSU140
      JK=IMM+1                                                          TIPSU141
      APHI=ATAN(PHI(IM+1))                                              TIPSU142
      SA=SIN(APHI)                                                      TIPSU143
      CA=COS(APHI)                                                      TIPSU144
      YLEG=Q(JK)+S(JK)*CA                                               TIPSU145
      YLEGSV(IUU,ITT) = YLEG                                            TIPSU146
      ZLEG=ZH(IM+1)+S(JK)*SA                                            TIPSU147
      ZLEGSV(IUU,ITT) = ZLEG                                            TIPSU148
      IF (XL(ITT).EQ.XT(ITT)) GO TO 150                                 TIPSU149
      TLX1=PN(JK)+S(JK)*TAN(PSI(JK))*CA                                 TIPSU150
      JK=JK+1                                                           TIPSU151
      TLX2=PN(JK)+S(JK)*TAN(PSI(JK))*CA                                 TIPSU152
      CRTTLG=TLX1-TLX2                                                  TIPSU153
      XTLEG(NA)=TLX1/2.+TLX2/2.                                         TIPSU154
      NAA=NA+1                                                          TIPSU155
      IF(ITT .EQ. 1) GO TO 130                                          TIPSU156
      LIM = ITT - 1                                                     TIPSU157
      DO 125 ITLEG = 1,LIM                                              TIPSU158
```

```
      L = NSSWSV(ITLEG) + 1                                             TIPSU159
      DO 120 I = 2,L                                                    TIPSU160
        IQ = I - 1                                                      TIPSU161
        IF((ABS(YLEGSV(I,ITLEG)-YLEG).LT.TOLC)                          TIPSU162
     $            .AND.                                                 TIPSU163
     $     (ABS(ZLEGSV(I,ITLEG)-ZLEG).LT.TOLC))                         TIPSU164
     $ CCIRS = CCIRS + CIRSUM(IQ,ITLEG)                                 TIPSU165
        IF(CCIRS .NE. 0)GO TO 125                                       TIPSU166
  120 CONTINUE                                                          TIPSU167
  125 CONTINUE                                                          TIPSU168
C                                                                       TIPSU169
 130  DO 140 NV=NAA,NB                                                  TIPSU170
      NVT=NV-1                                                          TIPSU171
 140  XTLEG(NV)=XTLEG(NVT)-CRTTLG                                       TIPSU172
      GO TO 40                                                          TIPSU173
C                                                                       TIPSU174
C                                                                       TIPSU175
 150  CONTINUE                                                          TIPSU176
      IF (CCIRS.NE.0.) GO TO 160                                        TIPSU177
      GO TO 270                                                         TIPSU178
 160  IJ=2*NSCW+1                                                       TIPSU179
      XLT=XTLEG(1)+CLFTLG/2.                                            TIPSU180
      XRT=XTLEG(IJ)+CRTTLG/2.                                           TIPSU181
      XLL=XLT+CLFTLG/4.                                                 TIPSU182
      XRL=XPT+CRTTLG/4.                                                 TIPSU183
      IF (XLL.GE.XL(ITT).AND.XLT.LE.XT(ITT)) GO TO 170                  TIPSU184
      IF (XLL.LE.XL(ITT).AND.XLT.GE.XT(ITT)) GO TO 190                  TIPSU185
      IF (XLL.GT.XL(ITT).AND.XLT.GE.XL(ITT)) GO TO 200                  TIPSU186
      IF (XLL.LE.XT(ITT)) GO TO 200                                     TIPSU187
      IF (XLL.GT.XL(ITT).AND.XLT.LT.XL(ITT)) GO TO 180                  TIPSU188
      CON4=(XT(ITT)-XLL)/(XLT-XLL)                                      TIPSU189
      GO TO 210                                                         TIPSU190
 170  CON4=(XL(ITT)-XT(ITT))/(XLL-XLT)                                  TIPSU191
      GO TO 210                                                         TIPSU192
 180  CON4=(XL(ITT)-XLT)/(XLL-XLT)                                      TIPSU193
      GO TO 210                                                         TIPSU194
 190  CON4=1.                                                           TIPSU195
      GO TO 210                                                         TIPSU196
 200  CON4=0.0                                                          TIPSU197
 210  CONTINUE                                                          TIPSU198
      IF (XRL.GE.XL(ITT).AND.XRT.LE.XT(ITT)) GO TO 220                  TIPSU199
      IF (XRL.LE.XL(ITT).AND.XRT.GE.XT(ITT)) GO TO 240                  TIPSU200
      IF (XRL.GT.XL(ITT).AND.XPT.GE.XL(ITT)) GO TO 250                  TIPSU201
      IF (XPL.LE.XT(ITT)) GO TO 250                                     TIPSU202
      IF (XRL.GT.XL(ITT).AND.XRT.LT.XL(ITT)) GO TO 230                  TIPSU203
      CON5=(XT(ITT)-XRL)/(XRT-XRL)                                      TIPSU204
      GO TO 260                                                         TIPSU205
 220  CON5=(XL(ITT)-XT(ITT))/(XRL-XRT)                                  TIPSU206
      GO TO 260                                                         TIPSU207
```

```
230   CON5=(XL(ITT)-XRT)/(XRL-XRT)                                    TIPSU208
      GO TO 260                                                       TIPSU209
240   CON5=1.                                                         TIPSU210
      GO TO 260                                                       TIPSU211
250   CON5=0.0                                                        TIPSU212
260   CONTINUE                                                        TIPSU213
      TIPSI = CCIRS*0.25*(CON4*WVOU(1)*CLFTLG-CON5*WVOU(IJ)*CRTTLG)   TIPSU214
     , *2./SREF*BETA                                                  TIPSU215
      IF(TIPSI .GT. 0 .AND. SSWWA(IM+1) .LT. 0) GO TO 270             TIPSU216
      IF(TIPSI .GT. 0)TIPSUM(ITT) = TIPSUM(ITT) + TIPSI               TIPSU217
      TIPSU = TIPSU + TIPSI                                           TIPSU218
      PITCH=PITCH+CCIRS*0.25*(-CON4*WVOU(1)*CLFTLG*XTLEG(1)+CON5*WVOU(IJTIPSU219
     1)*CRTTLG*XTLEG(IJ))*2./(SREF*CREF)*BETA**2                      TIPSU220
270   CIRCUS=CCIRS                                                    TIPSU221
      DO 460 NPOS=1,NSCW                                              TIPSU222
      JK=IMM+NPOS                                                     TIPSU223
      JN=2*NSCW+NPOS                                                  TIPSU224
      NPIS=NSCW+NPOS                                                  TIPSU225
      CIRCUS=CIRCUS+CIR(JK,2)                                         TIPSU226
      IF (XL(ITT).EQ.XT(ITT)) GO TO 460                               TIPSU227
      XLLEG=XTLEG(NPOS)                                               TIPSU228
      XRLEG=XTLEG(JN)                                                 TIPSU229
      XLL=XTLEG(NPOS)+CLFTLG/2.                                       TIPSU230
      XLT=XTLEG(NPOS)-CLFTLG/2.                                       TIPSU231
      XRL=XTLEG(JN)+CRTTLG/2.                                         TIPSU232
      XRT=XTLEG(JN)-CRTTLG/2.                                         TIPSU233
      IF (XLL.GE.XL(ITT).AND.XLT.LE.XT(ITT)) GO TO 280                TIPSU234
      IF (XLL.LE.XL(ITT).AND.XLT.GE.XT(ITT)) GO TO 300                TIPSU235
      IF (XLL.GT.XL(ITT).AND.XLT.GE.XL(ITT)) GO TO 310                TIPSU236
      IF (XLL.LE.XT(ITT)) GO TO 310                                   TIPSU237
      IF (XLL.GT.XL(ITT).AND.XLT.LT.XL(ITT)) GO TO 290                TIPSU238
      CON1=(XT(ITT)-XLL)/(XLT-XLL)                                    TIPSU239
      XLLEG=XT(ITT)+CON1*CLFTLG/2.                                    TIPSU240
      GO TO 320                                                       TIPSU241
280   CON1=(XL(ITT)-XT(ITT))/(XLL-XLT)                                TIPSU242
      XLLEG=(XL(ITT)+XT(ITT))/2.                                      TIPSU243
      GO TO 320                                                       TIPSU244
290   CON1=(XL(ITT)-XLT)/(XLL-XLT)                                    TIPSU245
      XLLEG=XLT+CON1*CLFTLG/2.                                        TIPSU246
      GO TO 320                                                       TIPSU247
300   CON1=1.                                                         TIPSU248
      GO TO 320                                                       TIPSU249
310   CON1=0.0                                                        TIPSU250
320   CONTINUE                                                        TIPSU251
      IF (NPOS.EQ.NSCW.AND.CON1.EQ.1.) GO TO 360                      TIPSU252
      IF (XRL.GE.XL(ITT).AND.XRT.LE.XT(ITT)) GO TO 330                TIPSU253
      IF (XRL.LE.XL(ITT).AND.XRT.GE.XT(ITT)) GO TO 350                TIPSU254
      IF (XRL.GT.XL(ITT).AND.XRT.GE.XL(ITT)) GO TO 370                TIPSU255
      IF (XRL.LE.XT(ITT)) GO TO 370                                   TIPSU256
```

```
      IF (XRL.GT.XL(ITT).AND.XRT.LT.XL(ITT)) GO TO 340                  TIPSU257
      CON2=(XT(ITT)-XRL)/(XRT-XRL)                                      TIPSU258
      XRLEG=XT(ITT)+CON2*CRTTLG/2.                                      TIPSU259
      GO TO 380                                                         TIPSU260
  330 CON2=(XL(ITT)-XT(ITT))/(XRL-XRT)                                  TIPSU261
      XRLEG=(XL(ITT)+XT(ITT))/2.                                        TIPSU262
      GO TO 380                                                         TIPSU263
  340 CON2=(XL(ITT)-XRT)/(XRL-XRT)                                      TIPSU264
      XRLEG=XRT+CON2*CRTTLG/2.                                          TIPSU265
      GO TO 380                                                         TIPSU266
  350 CON2=1.                                                           TIPSU267
      GO TO 380                                                         TIPSU268
  360 CON1=.75                                                          TIPSU269
      CON2=.75                                                          TIPSU270
      GO TO 380                                                         TIPSU271
  370 CON2=0.0                                                          TIPSU272
  380 IF (XRL.GT.XLL) GO TO 390                                         TIPSU273
      XSIGN=-1.0                                                        TIPSU274
      XBLL=XLL                                                          TIPSU275
      XBLT=XRL                                                          TIPSU276
      GO TO 400                                                         TIPSU277
  390 XBLL=XRL                                                          TIPSU278
      XBLT=XLL                                                          TIPSU279
      XSIGN=1.                                                          TIPSU280
  400 BVDLG=ABS(XBLL-XBLT)                                              TIPSU281
      IF (XBLT.GE.XL(ITT)) GO TO 440                                    TIPSU282
      IF (XBLL.LE.XT(ITT)) GO TO 440                                    TIPSU283
      IF (XBLL.GE.XL(ITT).AND.XBLT.LE.XT(ITT)) GO TO 430                TIPSU284
      IF (XBLL.LE.XL(ITT).AND.XBLT.GE.XT(ITT)) GO TO 420                TIPSU285
      IF (XBLL.GT.XL(ITT).AND.XBLT.GE.XT(ITT)) GO TO 410                TIPSU286
      CON3=(XT(ITT)-XBLL)/(XBLT-XBLL)                                   TIPSU287
      XTLEG(NPIS)=XT(ITT)+CON3*BVDLG/2.                                 TIPSU288
      CON3=CON3*XSIGN                                                   TIPSU289
      GO TO 450                                                         TIPSU290
  410 CON3=(XL(ITT)-XBLT)/(XBLL-XBLT)                                   TIPSU291
      XTLEG(NPIS)=XBLT+CON3*BVDLG/2.                                    TIPSU292
      CON3=CON3*XSIGN                                                   TIPSU293
      GO TO 450                                                         TIPSU294
  420 CON3=1.*XSIGN                                                     TIPSU295
      GO TO 450                                                         TIPSU296
  430 CON3=(XL(ITT)-XT(ITT))/(XBLL-XBLT)                                TIPSU297
      XTLEG(NPIS)=(XL(ITT)+XT(ITT))/2.                                  TIPSU298
      CON3=CON3*XSIGN                                                   TIPSU299
      GO TO 450                                                         TIPSU300
  440 CON3=0.0                                                          TIPSU301
  450 TIPSI =(CIRCUS*(WVOU(NPOS)*CLFTLG*CON1-CON2*WVOU(JN)*CRTTLG)      TIPSU302
     ,       +CIR(JK,2)*(WVOU(NPIS)*CON3*BVDLG))*2./SREF*BETA           TIPSU303
      IF (TIPSI .GT. 0 .AND. SSWWA(IM+1) .LT. 0) GO TO 460              TIPSU304
      IF(TIPSI .GT. 0)TIPSUM(ITT) = TIPSUM(ITT) + TIPSI                 TIPSU305
```

```
      TIPSU = TIPSU + TIPSI                                             TIPSU306
      PITCH=PITCH+(CIRCUS*(-WVOU(NPOS)*CLFTLG*CON1*XLLEG+WVOU(JN)*CON2*CTIPSU307
     1RTTLG*XRLEG)-CIR(JK,2)*(WVOU(NPIS)*CON3*BVDLG*XTLEG(NPIS)))*2./(SRTIPSU308
     2EF*CREF)*BETA**2                                                  TIPSU309
 460  CONTINUE                                                          TIPSU310
      IM=IM+1                                                           TIPSU311
      IMM=IMM+TBLSCW(IM)                                                TIPSU312
      IMP = IMP + 1                                                     TIPSU313
      CIRSUM(IMP,ITT) = CIRCUS - CCIRS                                  TIPSU314
      IUU = IMP + 2                                                     TIPSU315
  465 IF(IM .EQ. NSSW2)GO TO 475                                        TIPSU316
      IF(XL(ITT) .EQ. XT(ITT))GO TO 100                                 TIPSU317
C                                                                       TIPSU318
      IF(TBLSCW(IM+1) .NE. TBLSCW(IM))GO TO 471                         TIPSU319
      NCTL = 1                                                          TIPSU320
      CLFTLG = CRTTLG                                                   TIPSU321
      DO 470 NV = 1,NSCW                                                TIPSU322
        NY = NV + 2*NSCW                                                TIPSU323
        XTLEG(NV) = XTLEG(NY)                                           TIPSU324
        WVOU(NV) = WVOU(NY)                                             TIPSU325
  470 CONTINUE                                                          TIPSU326
      GO TO 80                                                          TIPSU327
  471 NSCW = TBLSCW(IM+1)                                               TIPSU328
      I = IMM + 1                                                       TIPSU329
      J = I + 1                                                         TIPSU330
      APHI = ATAN(PHI(IM+1))                                            TIPSU331
      GO TO 25                                                          TIPSU332
C                                                                       TIPSU333
C                                                                       TIPSU334
  475 CONTINUE                                                          TIPSU335
      KVSE(ITT) = 2.0 * ABS(TIPSU - CTSW)                               TIPSU336
      IF(KVSE(ITT) .LT. EPS)GO TO 510                                   TIPSU337
      CENTR(ITT) = (PITCH - CMW) * CREF/ABS(TIPSU - CTSW)               TIPSU338
C                                                                       TIPSU339
  510 CMW = PITCH                                                       TIPSU340
      CTSW = TIPSU                                                      TIPSU341
      NSSW1 = NSSWSV(ITT)                                               TIPSU342
C                                                                       TIPSU343
  535 CONTINUE                                                          TIPSU344
C                                                                       TIPSU345
C                                                                       TIPSU346
C                                                                       TIPSU347
  540 IF(CLDES .EQ. 100.) GO TO 541                                     TIPSU348
      CALL WRTANS(KVSE, CENTR, TIPSUM)                                  TIPSU349
      GO TO 575                                                         TIPSU350
  541 RAD=4.*ATAN(1.)/180.                                              TIPSU351
      WRITE(6,690)                                                      TIPSU352
      DO 580 IK=1,IPLAN                                                 TIPSU353
      WRITE(6,700) IK                                                   TIPSU354
```

```
      XL(IK) = XL(IK) * BETA                                            TIPSU355
      XT(IK) = XT(IK) * BETA                                            TIPSU356
      WRITE(6,710) XL(IK), XT(IK),                                      TIPSU357
     $             KVSE(IK), CENTR(IK), BOTSV(IK)                       TIPSU358
  580 CONTINUE                                                          TIPSU359
      WRITE(6,560)                                                      TIPSU360
      DO 600 IA=1,26                                                    TIPSU361
      READ(30) WVOU(IA),XTLEG(IA),CIRSUM(IA)                            TIPSU362
  600 CONTINUE                                                          TIPSU363
      DO 605 IA=1,26                                                    TIPSU364
      READ(30) YLEGSV(IA),ZLEGSV(IA),TBLSCW(IA)                         TIPSU365
  605 CONTINUE                                                          TIPSU366
      DO 610 IA=1,26                                                    TIPSU367
      READ(30) Q(IA),PN(IA),PV(IA)                                      TIPSU368
  610 CONTINUE                                                          TIPSU369
      DO 620 IA=1,26                                                    TIPSU370
      READ(30)ALP(IA),S(IA),PSI(IA)                                     TIPSU371
  620 CONTINUE                                                          TIPSU372
      REWIND 30                                                         TIPSU373
      CLOZ = 0.0                                                        TIPSU374
      CDMINZ = XTLEG(1)                                                 TIPSU375
      CLOF = 0.0                                                        TIPSU376
      CDMINF = ZLEGSV(1)                                                TIPSU377
      CDMINV = 100.0                                                    TIPSU378
      CLOV = 0.0                                                        TIPSU379
      DO 545 IALPH = 1,26                                               TIPSU380
      ALPH(IALPH) = (IALPH-6)*2.                                        TIPSU381
      ALPHA = ALPH(IALPH)*RAD                                           TIPSU382
      SA=SIN(ALPHA)                                                     TIPSU383
      SA2=SA*ABS(SA)                                                    TIPSU384
      CA=COS(ALPHA)                                                     TIPSU385
      CLVSE=CDVSE=CMVSE=0.                                              TIPSU386
      DO 550 I=1,IPLAN                                                  TIPSU387
      CLVSE = KVSE(I) * SA2 * CA + CLVSE                                TIPSU388
      CDVSE = KVSE(I) * SA2 * SA + CDVSE                                TIPSU389
      CMVSE = KVSE(I) * SA2 * CENTR(I)/CREF + CMVSE                     TIPSU390
  550 CONTINUE                                                          TIPSU391
      WRITE(6,570) ALPH(IALPH),WVOU(IALPH), XTLEG(IALPH), CIRSUM(IALPH),TIPSU392
     1             YLEGSV(IALPH), ZLEGSV(IALPH), TBLSCW(IALPH),         TIPSU393
     2             Q(IALPH),PN(IALPH),PV(IALPH),CLVSE,CDVSE,CMVSE       TIPSU394
      IF (XTLEG(IALPH).GE.CDMINZ) GO TO 900                             TIPSU395
      CDMINZ = XTLEG(IALPH)                                             TIPSU396
      CLOZ = WVOU(IALPH)                                                TIPSU397
  900 CONTINUE                                                          TIPSU398
      IF (ZLEGSV(IALPH).GE.CDMINF) GO TO 910                            TIPSU399
      CDMINF = ZLEGSV(IALPH)                                            TIPSU400
      CLOF = YLEGSV(IALPH)                                              TIPSU401
  910 CONTINUE                                                          TIPSU402
      WRITE(30) CLVSE,CDVSE,CMVSE                                       TIPSU403
```

```
  545 CONTINUE                                                          TIPSU404
      END FILE 30                                                       TIPSU405
      REWIND 30                                                         TIPSU406
      WRITE(6,800)                                                      TIPSU407
      DO 810 IALPH=1,26                                                 TIPSU408
      READ (30) CLVSE,CDVSE,CMVSE                                       TIPSU409
      CLL=Q(IALPH)+CLVSE                                                TIPSU410
      CDD=PN(IALPH)+CDVSE                                               TIPSU411
      CMM=PV(IALPH)+CMVSE                                               TIPSU412
      CLLA=Q(IALPH)+ALP(IALPH)                                          TIPSU413
      CDDA=PN(IALPH)+S(IALPH)                                           TIPSU414
      CMMA=PV(IALPH)+PSI(IALPH)                                         TIPSU415
      CLLL(IALPH) = CLL+ALP(IALPH)                                      TIPSU416
      CDDD(IALPH) = CDD+S(IALPH)                                        TIPSU417
      CMMM(IALPH)= CMM+PSI(IALPH)                                       TIPSU418
      WRITE(6,570) ALPH(IALPH),ALP(IALPH),S(IALPH),PSI(IALPH),CLL,CDD,  TIPSU419
     1CMM,CLLA,CDDA,CMMA,CLLL(IALPH), CDDD(IALPH), CMMM(IALPH)          TIPSU420
      IF (CDDD(IALPH).GE.CDMINV) GO TO 920                              TIPSU421
      CDMINV = CDDD(IALPH)                                              TIPSU422
      CLOV = CLLL(IALPH)                                                TIPSU423
  920 CONTINUE                                                          TIPSU424
  810 CONTINUE                                                          TIPSU425
      PIAR = 1./(3.14159265*AR)                                         TIPSU426
      WRITE (6,831) PIAR,CLOZ,CDMINZ,CLOF,CDMINF,CLOV,CDMINV            TIPSU427
      DO 820 IALPH=1,26                                                 TIPSU428
      IF(WVOU(IALPH).EQ.CLOZ) CDCLZ(IALPH) = 100.                       TIPSU429
      IF(YLEGSV(IALPH).EQ.CLOF) CDCLF(IALPH) = 100.                     TIPSU430
      IF(CLLL(IALPH).EQ.CLOV) CDCLV(IALPH) = 100.                       TIPSU431
      IF(WVOU(IALPH).NE.CLOZ)CDCLZ(IALPH) = (XTLEG(IALPH)-CDMINZ)/((WVOUTIPSU432
     1 (IALPH) - CLOZ)**2)                                              TIPSU433
      IF(YLEGSV(IALPH).NE.CLOF) CDCLF(IALPH)= (ZLEGSV(IALPH)-CDMINF)/   TIPSU434
     1  ((YLEGSV(IALPH)- CLOF)**2)                                      TIPSU435
      IF(CLLL(IALPH).NE.CLOV) CDCLV(IALPH)= (CDDD(IALPH)- CDMINV)/      TIPSU436
     1 ((CLLL(IALPH)- CLOV)**2)                                         TIPSU437
      WRITE(6,830) ALPH(IALPH), WVOU(IALPH),CDCLZ(IALPH),YLEGSV(IALPH), TIPSU438
     1 CDCLF(IALPH), CLLL(IALPH), CDCLV(IALPH)                          TIPSU439
  820 CONTINUE                                                          TIPSU440
      CALL PSEUDO                                                       TIPSU441
C                                                                       TIPSU442
C     PLOT  CL VS ALPHA  FOR ZERO SUCTION, FULL SUCTION, AND VORTEX LIFTTIPSU443
C                                                                       TIPSU444
      CALL INFOPLT( 0,26, ALPH ,1,YLEGSV,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU445
     15HALPHA,-2, 2HCL,12,10.,10.,0.75,1.50)                            TIPSU446
      CALL INFOPLT( 0,26, ALPH ,1,YLEGSV,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU447
     15HALPHA,-2, 2HCL, 0,10.,10.,0.75,1.50)                            TIPSU448
      CALL INFOPLT( 0,26, ALPH ,1, WVOU ,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU449
     15HALPHA,-2, 2HCL, 0,10.,10.,0.75,1.50)                            TIPSU450
      CALL INFOPLT( 0,26, ALPH ,1, WVOU ,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU451
     15HALPHA,-2, 2HCL,11,10.,10.,0.75,1.50)                            TIPSU452
```

```
      CALL INFOPLT( 0,26, ALPH ,1, CLLL ,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU453
     15HALPHA,-2, 2HCL, 0,10.,10.,0.75,1.50)                            TIPSU454
      CALL NOTATE ( 7.0, 2.7,0.15,12HZERO SUCTION,0.0,12)               TIPSU455
      CALL PNTPLT ( 6.7,2.77,11,2)                                      TIPSU456
      CALL NOTATE ( 7.0, 2.4,0.15,12HFULL SUCTION,0.0,12)               TIPSU457
      CALL PNTPLT ( 6.7,2.47,12,2)                                      TIPSU458
      CALL NOTATE ( 7.0, 2.1,0.15,16HPOTENTIAL+VORTEX,0.0,16)           TIPSU459
      CALL PNTPLT ( 6.7,2.17,13,2)                                      TIPSU460
      CALL INFOPLT( 1,26, ALPH ,1, CLLL ,1,-10., 40.0,-.4,1.6,1.0,-5,   TIPSU461
     15HALPHA,-2, 2HCL,13,10.,10.,0.75,1.50)                            TIPSU462
C                                                                       TIPSU463
C     PLOT  CL VS CD      FOR ZERO SUCTION, FULL SUCTION, AND VORTEX LIFTTIPSU464
C                                                                       TIPSU465
      CALL INFOPLT(0,26,WVOU,1,XTLEG,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,     TIPSU466
     1             2HCD, 0,10.,10.,0.75,1.5)                            TIPSU467
      CALL INFOPLT(0,26,YLEGSV,1,ZLEGSV,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,  TIPSU468
     1             2HCD, 0,10.,10.,0.75,1.5)                            TIPSU469
      CALL INFOPLT( 0,26,CLLL,1,CDDD,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,     TIPSU470
     1             2HCD, 0,10.,10.,0.75,1.5)                            TIPSU471
      CALL INFOPLT(0,26,WVOU,1,XTLEG,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,     TIPSU472
     1             2HCD,11,10.,10.,0.75,1.5)                            TIPSU473
      CALL INFOPLT(0,26,YLEGSV,1,ZLEGSV,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,  TIPSU474
     1             2HCD,12,10.,10.,0.75,1.5)                            TIPSU475
      CALL NOTATE ( 7.0, 2.7,0.15,12HZERO SUCTION,0.0,12)               TIPSU476
      CALL PNTPLT ( 6.7,2.77,11,2)                                      TIPSU477
      CALL NOTATE ( 7.0, 2.4,0.15,12HFULL SUCTION,0.0,12)               TIPSU478
      CALL PNTPLT ( 6.7,2.47,12,2)                                      TIPSU479
      CALL NOTATE ( 7.0, 2.1,0.15,16HPOTENTIAL+VORTEX,0.0,16)           TIPSU480
      CALL PNTPLT ( 6.7,2.17,13,2)                                      TIPSU481
      CALL INFOPLT( 1,26,CLLL,1,CDDD,1,-.4,1.6,0,.4,1.0,-2,2HCL,-2,     TIPSU482
     1             2HCD,13,10.,10.,0.75,1.5)                            TIPSU483
C                                                                       TIPSU484
C     PLOT  CL VS CM      FOR ZERO SUCTION, FULL SUCTION, AND VORTEX LIFTTIPSU485
C                                                                       TIPSU486
      CALL INFOPLT( 0,26,YLEGSV,1,TBLSCW,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU487
     1  2HCL,-2, 2HCM,12,10.,10.,0.75,1.50)                             TIPSU488
      CALL INFOPLT( 0,26,YLEGSV,1,TBLSCW,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU489
     1  2HCL,-2, 2HCM, 0,10.,10.,0.75,1.50)                             TIPSU490
      CALL INFOPLT( 0,26, WVOU ,1,CIRSUM,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU491
     1  2HCL,-2, 2HCM,11,10.,10.,0.75,1.50)                             TIPSU492
      CALL INFOPLT( 0,26, WVOU ,1,CIRSUM,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU493
     1  2HCL,-2, 2HCM, 0,10.,10.,0.75,1.50)                             TIPSU494
      CALL INFOPLT( 0,26, CLLL ,1, CMMM ,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU495
     1  2HCL,-2, 2HCM, 0,10.,10.,0.75,1.50)                             TIPSU496
      CALL NOTATE ( 7.0, 2.7,0.15,12HZERO SUCTION,0.0,12)               TIPSU497
      CALL PNTPLT ( 6.7,2.77,11,2)                                      TIPSU498
      CALL NOTATE ( 7.0, 2.4,0.15,12HFULL SUCTION,0.0,12)               TIPSU499
      CALL PNTPLT ( 6.7,2.47,12,2)                                      TIPSU500
      CALL NOTATE ( 7.0, 2.1,0.15,16HPOTENTIAL+VORTEX,0.0,16)           TIPSU501
```

```
      CALL PNTPLT ( 6.7,2.17,13,2)                                      TIPSU502
      CALL INFOPLT( 1,26, CLLL ,1, CMMM ,1,-.4,1.6,-0.10,0.10,1.0,-2,   TIPSU503
     1  2HCL,-2, 2HCM,13,10.,10.,0.75,1.50)                             TIPSU504
C                                                                       TIPSU505
C     PLOT  CL VS CD/(CL**2) FOR ZERO SUCTION, FULL SUCTION, + VORTEX CLTIPSU506
C                                                                       TIPSU507
      CALL INFOPLT( 0,26,YLEGSV,1, CDCLF,1,-.4,1.6,0.0,1.0,1.0,-2,      TIPSU508
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2), 0,10.,10.,0.75,1.5)         TIPSU509
      CALL INFOPLT( 0,26,YLEGSV,1, CDCLF,1,-.4,1.6,0.0,1.0,1.0,-2,      TIPSU510
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2),12,10.,10.,0.75,1.5)         TIPSU511
      CALL INFOPLT( 0,26, WVOU ,1,CDCLZ,1, -.4, 1.6, 0.0, 1.0, 1.0,-2,  TIPSU512
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2), 0,10.,10.,0.75,1.5)         TIPSU513
      CALL INFOPLT( 0,26, WVOU ,1,CDCLZ,1, -.4, 1.6, 0.0, 1.0, 1.0,-2,  TIPSU514
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2),11,10.,10.,0.75,1.5)         TIPSU515
      CALL INFOPLT( 0,26, CLLL ,1,CDCLV,1, -.4, 1.6, 0.0, 1.0,1.0,-2,   TIPSU516
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2), 0,10.,10.,0.75,1.5)         TIPSU517
      CALL NOTATE ( 7.0, 2.7,0.15,12HZERO SUCTION,0.0,12)               TIPSU518
      CALL PNTPLT ( 6.7,2.77,11,2)                                      TIPSU519
      CALL NOTATE ( 7.0, 2.4,0.15,12HFULL SUCTION,0.0,12)               TIPSU520
      CALL PNTPLT ( 6.7,2.47,12,2)                                      TIPSU521
      CALL NOTATE ( 7.0, 2.1,0.15,16HPOTENTIAL+VORTEX,0.0,16)           TIPSU522
      CALL PNTPLT ( 6.7,2.17,13,2)                                      TIPSU523
      CALL INFOPLT( 1,26, CLLL ,1,CDCLV,1, -.4, 1.6, 0.0, 1.0,1.0,-2,   TIPSU524
     12HCL,-24,24H(CD-COMIN)/((CL-CLO)**2),13,10.,10.,0.75,1.5)         TIPSU525
      CALL CALPLT(0,0,999)                                              TIPSU526
  830 FORMAT(7F15.5)                                                    TIPSU527
  831 FORMAT(1H1///20X66H I N D U C E D    D R A G    P O L A R    S H ATIPSU528
     1 P E     F A C T O R//45X12H1/(PI*AR) = ,F 8.5,//27X               TIPSU529
     1 15HZERO LE SUCTION, 15X15HFULL LE SUCTION,                       TIPSU530
     2  10X25HPOTENT.+VORTEX(LE+SE+AUG)// 26X3HCLO, 8X5HCDMIN, 14X3HCLO,TIPSU531
     3  8X5HCDMIN, 14X3HCLO, 8X5HCDMIN/15X,6F15.5 //9X5HALPHA,12X2HCL,9XTIPSU532
     411H(CD-COMIN)/  8X2HCL,9X11H(CD-COMIN)/  8X2HCL, 9X11H(CD-COMIN)/,TIPSU533
     5 / 38X13H((CL-CLO)**2), 16X13H((CL-CLO)**2),16X13H((CL-CLO)**2 )  TIPSU534
  560 FORMAT(1H1,//,47X,33HTOTAL PERFURMANCE CHARACTERISTICS,//         TIPSU535
     110X,110HA    T    T    A    C    H    E    D    F    L    O       TIPSU536
     2 W          S  E  P  A  R  A  T  E  D      F  L  O  W/            TIPSU537
     372X28HPLUS POTENTIAL CONTRIBUTIONS/                 13X112HZERO   TIPSU538
     4LEADING  EDGE  SUCTION  FULL  LEADING  EDGE  SUCTION     L E A D ITIPSU539
     5 N G     E D G E        S I D E    E D G E/3X,127HALPHA       CL   TIPSU540
     6     CD          CM          CL          CD          CM        CL CTIPSU541
     7D         CM         CL VSE      CD VSE     CM VSE)               TIPSU542
  570 FORMAT(13F10.5)                                                   TIPSU543
  690 FORMAT (1H1,////,31X,60H KVSE  AND  RESPECTIVE CHORDWISE CENTROID TIPSU544
     1 FOR EACH PLANFORM)                                               TIPSU545
C                                                                       TIPSU546
  700 FORMAT(//,60X,14HPLANFORM  NO.  ,I2,/,                            TIPSU547
     $ 90X, 8HLOCATION,/,                                               TIPSU548
     $ 33X, 21HLIMITS OF INTEGRATION, 18X, 5HKV SE,                     TIPSU549
     $ 5X, 9HCHORDWISE, 3X, 8HSPANWISE)                                 TIPSU550
```

```
C                                                                           TIPSU551
  710 FORMAT(23X, F10.5, 10H (LEADING), 2X, F10.5, 11H (TRAILING),          TIPSU552
     $ 4X, F10.5, 2X, F10.5, 2X, F10.5)                                     TIPSU553
C                                                                           TIPSU554
  800 FORMAT(1H1///4X118HS     E     P     A     R     A     T     E     D  TIPSU555
     1    F     L     O     W          C     O     N     T     I     N     U     ETIPSU556
     2    D/45X80HP   L  U  S     P  O  T  E  N  T  I  A  L     C  O  N  T TIPSU557
     3 R  I  B  U  T  I  O  N  S/15X109HA U G M E N T E D          LEADTIPSU558
     4ING EDGE  +  SIDE EDGE    LEADING EDGE  +  AUGMENTED       LE + SE TIPSU559
     5+ AUGMENTED/                                                          TIPSU560
     6                                 3X125HALPHA        CL AUG     CD ATIPSU561
     7UG     CM AUG       CL          CD          CM          CL          CD   TIPSU562
     8  CM          CL          CD          CM)                             TIPSU563
  575 CONTINUE                                                              TIPSU564
      END                                                                   TIPSU565
      SUBROUTINE WRTANS(KVSE, CENTR, TIPSUM)                                WRTANS 2
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                    ALL    2
     $    STA(4), TBLSCW(100), YYCP(4),                                     ALL    3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),             ALL    4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                            ALL    5
C                                                                           ALL    6
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,        ONETHRE2
     $   ARTRUE, RTCOHT(4), CONFIG(2), NSSWSV(4),                           ONETHRE3
     $   MSV(4), KBOT, PLAN, IPLAN, MACH,                                   ONETHRE4
     $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),          ONETHRE5
     $  CLAMAP(4), CLWIN(4), CLWNG(4), XLOCTN,                              ONETHRE6
     $  YINNER(4), YOUTER(4)                                                ONETHRE7
C                                                                           ONETHRE8
      INTEGER CONFIG                                                        ONETHRE9
C                                                                           ONETHR10
      COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD                   THREFOR2
C                                                                           THREFOR3
      REAL KP(4), KVLE(4), KVSE(4)                                          WRTANS 6
      DIMENSION CENTR(4), CENT(4), TIPSUM(4), CENTY(4)                      WRTANS 7
      LCH=0                                                                 WRTANS 8
      LAMAR=NSSWSV(1)                                                       WRTANS 9
      CONV=3.1415926536/180.                                                WRTANS10
      CINV=1./(3.1415926536*AR)                                             WRTANS11
      DELTA=2.*CONV                                                         WRTANS12
      CONST=16.*BOT/SREF                                                    WRTANS13
      ALPHA=ALPD*CONV                                                       WRTANS14
      S22=ALPHA**2                                                          WRTANS15
      EPS = 1.E-6                                                           WRTANS16
      DO 10 ITT=1,IPLAN                                                     WRTANS17
        CENT(ITT) = 0.0                                                     WRTANS18
        XT(ITT) = XT(ITT) * BETA                                            WRTANS19
        XL(ITT) = XL(ITT) * BETA                                            WRTANS20
        CENTY(ITT) = 0.0                                                    WRTANS21
        KP(ITT) = 0.0                                                       WRTANS22
```

```
        KVLE(ITT) = 0.0
   10 CONTINUE
C
      KP(1) = CLWIN(1)
      IF(IPLAN .EQ. 1) GO TO 20
      DO 15 ITT = 2,IPLAN
        KP(ITT) = CLWIN(ITT) - CLWIN(ITT-1)
   15 CONTINUE
   20 CONTINUE
C
C
C
      IEDGE = 1
      INDEX = 0
      DO 50 ITT = 1,IPLAN
        MSPAN = NSSWSV(ITT)
        PITCHMO = 0.0
        FORCE = 0.0
        ROLLMO = 0.0
        DO 30 ISPAN = 1,MSPAN
          INDEX = INDEX + 1
          IF(Q(IEDGE) .GT. YINNER(ITT)) GO TO 25
          IF(Q(IEDGE) .LT. YOUTER(ITT)) GO TO 25
          PITCHMO = PITCHMO + (PN(IEDGE) +
     $       (PN(IEDGE)-PN(IEDGE+1))/4.0)* BETA *
     $       BLAIR(INDEX) * S(IEDGE) * CONST
          FORCE = FORCE + BLAIR(INDEX) * S(IEDGE) * CONST
          ROLLMO = ROLLMO + Q(IEDGE) * BLAIR(INDEX) *
     $       S(IEDGE) * CONST
   25     IEDGE = IEDGE + TBLSCW(INDEX)
   30   CONTINUE
        IF(FORCE .EQ. 0.0) GO TO 50
        KVLE(ITT) = FORCE / S22
        IF(PITCHMO .NE. 0.0) CENT(ITT) = PITCHMO / FORCE
        IF(ROLLMO  .NE. 0.0) CENTY(ITT) = ROLLMO/ FORCE/ BOT
        IF(KVLE(ITT) .GT. EPS) GO TO 50
        CENT(ITT) = 0.0
        CENTY(ITT) = 0.0
   50 CONTINUE
C
100   CONTINUE
      WRITE (6,190)
      DO 110 IK=1,IPLAN
      CENTPM=CLAMAR(IK)*CREF
      WRITE (6,200) IK
      SPANOFK = YYCP(IK) * BOT
      WRITE(6,210) KP(IK), CENTPM, SPANOFK
      CENTYSP = CENTY(IK) * BOT
      WRITE(6,220) YINNER(IK), YOUTER(IK),
```

```
     $              KVLE(IK), CENT(IK), CENTYSP                         WRTANS72
      WRITE(6,230) XL(IK), XT(IK),                                      WRTANS73
     $              KVSE(IK), CENTR(IK), BOTSV(IK)                      WRTANS74
      WRITE(6,290) TIPSUM(IK)                                           WRTANS75
  110 CONTINUE                                                          WRTANS76
120   CONTINUE                                                          WRTANS77
      DO 160 IK=1,IPLAN                                                 WRTANS78
      IF (LCH.EQ.1) GO TO 130                                           WRTANS79
      WRITE (6,240) IK                                                  WRTANS80
130   WRITE (6,250)                                                     WRTANS81
      ALPHA=0.0                                                         WRTANS82
      DO 160 J=1,26                                                     WRTANS83
      V=SIN(ALPHA)                                                      WRTANS84
      C=COS(ALPHA)                                                      WRTANS85
      C2=C**2                                                           WRTANS86
      S2=V**2                                                           WRTANS87
      IF(LCH .EQ. 1) GO TO 140                                          WRTANS88
C         INDIVIDUAL PLANFORM CHARACTERISTICS                           WRTANS89
C                                                                       WRTANS90
      CLP = KP(IK) * V * C2                                             WRTANS91
      CLVL = CLP + KVLE(IK) * S2 * C                                    WRTANS92
      CLSL = CLP + KVSE(IK) * S2 * C                                    WRTANS93
      CLTOT = CLVL + KVSE(IK) * S2 * C                                  WRTANS94
      CMP = CLAMAR(IK) * KP(IK) * V * C                                 WRTANS95
      CMPL  = CMP + CENT(IK) * KVLE(IK) * S2/CREF                       WRTANS96
      CMPS  = CMP + KVSE(IK) * CENTR(IK) * S2/CREF                      WRTANS97
      CMTOT = CMPL + KVSE(IK) * CENTR(IK) * S2/CREF                     WRTANS98
      GO TO 150                                                         WRTANS99
C                                                                       WRTAN100
C         TOTAL PLANFORM CHARACTERISTICS                                WRTAN101
  140 CLP = SKP * V * C2                                                WRTAN102
      CLVL = CLP + SKVLE * S2 * C                                       WRTAN103
      CLSL = CLP + SKVSE * S2 * C                                       WRTAN104
      CLTOT = CLVL + SKVSE * S2 * C                                     WRTAN105
      CMP = V * C * SCMP                                                WRTAN106
      CMPL = CMP + S2 * SCMPL/CREF                                      WRTAN107
      CMPS = CMP + S2 * SCMPS/CREF                                      WRTAN108
      CMTOT = CMPL + CMPS - CMP                                         WRTAN109
C                                                                       WRTAN110
150   CDI=CLTOT*TAN(ALPHA)                                              WRTAN111
      CDII=(CLTOT**2)*CINV                                              WRTAN112
      ALPH1=ALPHA/CONV                                                  WRTAN113
      CNTT=CLTOT/C                                                      WRTAN114
      WRITE (6,260) ALPH1,CNTT,CLP,CLVL,CLSL,CLTOT,CMP,CMPL,CMPS,CMTOT,CWRTAN115
     1DI,CDII                                                           WRTAN116
160   ALPHA=ALPHA+DELTA                                                 WRTAN117
      IF (IPLAN.EQ.1) GO TO 170                                         WRTAN118
      IPLAN=1                                                           WRTAN119
      LCH=1                                                             WRTAN120
```

```
      WRITE (6,280)                                                     WRTAN121
      SKP = 0.0                                                         WRTAN122
      SKVLE = 0.0                                                       WRTAN123
      SKVSE = 0.0                                                       WRTAN124
      SCMP = 0.0                                                        WRTAN125
      SCMPL = 0.0                                                       WRTAN126
      SCMPS = 0.0                                                       WRTAN127
C                                                                       WRTAN128
      K = IFIX(PLAN)                                                    WRTAN129
      DO 165 ITT = 1,K                                                  WRTAN130
        SKP = SKP + KP(ITT)                                             WRTAN131
        SKVLE = SKVLE + KVLE(ITT)                                       WRTAN132
        SKVSE = SKVSE + KVSE(ITT)                                       WRTAN133
        SCMP = SCMP + KP(ITT) * CLAMAR(ITT)                             WRTAN134
        SCMPL = SCMPL + KVLE(ITT) * CENT(ITT)                           WRTAN135
        SCMPS = SCMPS + KVSE(ITT) * CENTR(ITT)                          WRTAN136
  165 CONTINUE                                                          WRTAN137
C                                                                       WRTAN138
C                                                                       WRTAN139
      GO TO 120                                                         WRTAN140
170   WRITE (6,270)                                                     WRTAN141
      IPLAN=PLAN                                                        WRTAN142
      RETURN                                                            WRTAN143
C                                                                       WRTAN144
C                                                                       WRTAN145
C                                                                       WRTAN146
  180 FORMAT(8F10.5)                                                    WRTAN147
  190 FORMAT(1H1,//,41X,                                                WRTAN148
     $ 50HKP , KV AND RESPECTIVE CENTROIDS FOR EACH PLANFORM  )         WRTAN149
C                                                                       WRTAN150
  200 FORMAT(//,55X,18HPLANFORM  NUMBER   ,I2,/,                        WRTAN151
     $  92X,8HLOCATION,/,                                               WRTAN152
     $  77X,5HVALUE,3X,9HCHORDWISE,4X,8HSPANWISE,/,                     WRTAN153
     $  21X,21HLIMITS OF INTEGRATION )                                  WRTAN154
C                                                                       WRTAN155
  210 FORMAT(65X,7HKP     ,3(F10.5,2X))                                 WRTAN156
C                                                                       WRTAN157
  220 FORMAT(10X,F10.5,2X,7H(INNER),6X,F10.5,2X,7H(OUTER),              WRTAN158
     $ 11X,7HKV LE   ,3(F10.5,2X))                                      WRTAN159
C                                                                       WRTAN160
  230 FORMAT(10X,F10.5,2X,9H(LEADING),4X,F10.5,2X,10H(TRAILING) ,       WRTAN161
     $ 8X,7HKV SE   ,3(F10.5,2X))                                       WRTAN162
C                                                                       WRTAN163
240    FORMAT (1H1,////,43X,40HPERFORMANCE CHARACTERISTICS FOR PLANFORM,IWRTAN164
     12)                                                                WRTAN165
250    FORMAT (//7X,5HALPHA,6X,2HCN,8X,3HCLP,4X,9HCLP+CLVLE,1X,9HCLP+CLVSWRTAN166
     1E,4X,2HCL,8X,3HCMP,4X,9HCMP+CMVLE,1X,9HCMP+CMVSE,4X,2HCM,8X,2HCD,3WRTAN167
     2X,13HCL**2/(PI*AR)/)                                              WRTAN168
  260 FORMAT(3X,12F10.5)                                                WRTAN169
```

```
270   FORMAT (///,50X,21HTHIS CASE IS FINISHED)                                WRTAN170
280   FORMAT (1H1,////,48X,33HTOTAL PERFORMANCE CHARACTERISTICS)                WRTAN171
  290 FORMAT(1H0,32X,                                                           WRTAN172
     $ 47HSUM OF THE POSITIVE SIDE EDGE CONTRIBUTIONS =  ,F10.5)               WRTAN173
C                                                                               WRTAN174
      END                                                                       WRTAN175
      SUBROUTINE VORTEX                                                         VORTEX 2
      DIMENSION GAM(1000),XC4(1000),YQ(1000),CCR(40),                           VORTEX 3
     $ FW(2),FV(2),XXCC(400),CCC(400),CCU(400),YB(100),                         VORTEX 4
     $ BOTL(4), NUMBER(4), NMASUM(4),                                           VORTEX 5
     $ CRI(102),NMA(4),XCC4(400),CHD(100),XC44(100),YY(2),                      VORTEX 6
     $ PPHI(100),ZZH(100),Z(1000),PHII(1000),SA(100),                           VORTEX 7
     $ SSA(1000),ALOP(400),ALLP(100),ALPPD(1000),ALO(40),                       VORTEX 8
     $ YC(102),YQQ(100),CCRR(40),CRJ(102),YCHLO(4),YCHHI(4),                    VORTEX 9
     $ VELIN(26),CDRAGIT(26),CLL(26),GAD(1000),CUTE(26),                        VORTEX10
     $ CMM(26),CDRAG(26),CLIFT(26),CPITCH(26),                                  VORTEX11
     $ CSS(26),CSUCT(26),CLAUG(26),CDAUG(26),CMAUG(26),CLV(26)                  VORTEX12
C                                                                               VORTEX13
      COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,                        ALL    2
     $    STA(4), TBLSCW(100), YYCP(4),                                         ALL    3
     $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),                 ALL    4
     $    PHI(100), ZH(100), CP(400), STLOIND(4)                                ALL    5
C                                                                               ALL    6
C                                                                               VORTEX15
C                                                                               NOTES  2
C             ---  NOTES TO THE USERS  ---                                      NOTES  3
C                                                                               NOTES  4
C                                                                               NOTES  5
C        1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING                       NOTES  6
C           EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF                      NOTES  7
C           ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE                         NOTES  8
C           CALCULATED FROM CLDES = 100. AND CLDES = 1.0                        NOTES  9
C                                                                               NOTES 10
C                                                                               NOTES 11
C         2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOM-           NOTES 12
C            MENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE          NOTES 13
C           MORE MEANINGFUL SIDE EDGE RESULTS                                   NOTES 14
C                                                                               NOTES 15
C                                                                               NOTES 16
C                                                                               NOTES 17
C         3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO                NOTES 18
C            0. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE                NOTES 19
C            EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO              NOTES 20
C            1.0 IF THIS LOADING IS TO BE NON-ZERO                              NOTES 21
C                                                                               NOTES 22
C                                                                               NOTES 23
C                                                                               VORTEX17
      COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,            ONETHRE2
     $   ARTRUE, RTCOHT(4), CONFIG(2), NSSWSV(4),                               ONETHRE3
     $   MSV(4), KPOT, PLAN, IPLAN, MACH,                                       ONETHRE4
     $   SSWWA(100), XL(4), YT(4), CLWB, CMCL, CIA(4), BLAIR(100),              ONETHRE5
     $  CLAMAR(4), CLWIN(4), CLWNC(4), YLOCTN,                                   ONETHRE6
     $  YINNER(4), YOUTER(4)                                                    ONETHRE7
C                                                                               ONETHRE8
      INTEGER CONFIG                                                            ONETHRE9
```

```
C                                                                       ONETHR10
      COMMON /TOTHREE/ CIR(400,2)                                       TOTHRE 2
C                                                                       TOTHRE 3
      COMMON /INSUB23/ APSI,APHI,XX,YYY,ZZ,SNN,TOLCSQ                   VORTEX20
      COMMON /THREEFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD              THREEFOR2
C                                                                       THREEFOR3
      COMMON/CORROD/ TSPAN(4), TSPANA, KBIT, CTILDA, XTILDA, DISTALE    CORROD 2
C                                                                       CORROD 3
C                                                                       VORTEX23
      DIMENSION CDG(26,4), CLF(26,4), CPH(26,4), CST(26,4)              VORTEX24
C                                                                       VORTEX25
      DATA NUMBER/6HFIRST ,6HSECOND,6HTHIRD ,6HFOURTH/                  VORTEX26
C                                                                       VORTEX27
C                                                                       VORTEX30
      ICOUNT=1                                                          VORTEX31
      DO 1 ITT = 1,IPLAN                                                VORTEX32
      YCHHI(ITT) = -0.0                                                 VORTEX33
      YCHLO(ITT) = -0.0                                                 VORTEX34
    1 CONTINUE                                                          VORTEX35
      REWIND 30                                                         VORTEX36
C                                                                       VORTEX37
    5 WRITE (6,250)                                                     VORTEX38
      DO 10 I=1,IPLAN                                                   VORTEX39
   10 WRITE(6,255) I,YCHLO(I),YCHHI(I)                                  VORTEX40
      IF(ICOUNT.EQ.1) WRITE(6,275)                                      VORTEX41
      IF(ICOUNT.EQ.2) WRITE(6,280)                                      VORTEX42
      REWIND 20                                                         VORTEX43
      APSI=TOLCSQ=TBLS=0.                                               VORTEX44
      PI=4.*ATAN(1.)                                                    VORTEX45
      FPI=4.*PI                                                         VORTEX46
      DO 15 ITT = 1,IPLAN                                               VORTEX47
        BOTL(ITT) = ABS(TSPAN(ITT))                                     VORTEX48
   15 CONTINUE                                                          VORTEX49
C                                                                       VORTEX50
      SNN = BOTL(KBOT) / (2.0 * NSSWSV(KBOT))                           VORTEX51
      DELTYB=2.*SNN                                                     VORTEX52
      NMAX = 0                                                          VORTEX53
      DO 17 ITT=1,IPLAN                                                 VORTEX54
      NMA(ITT) = BOTL(ITT) / DELTYB                                     VORTEX55
        IF (NMA(ITT) .EQ. 0) GO TO 18                                   VORTEX56
        NMAX= NMAX + NMA(ITT)                                           VORTEX57
        NMASUM(ITT) = NMAX                                              VORTEX58
   17 CONTINUE                                                          VORTEX59
      GO TO 19                                                          VORTEX60
   18 WRITE(6,290)                                                      VORTEX61
      M = -1                                                            VORTEX62
      GO TO 400                                                         VORTEX63
C                                                                       VORTEX64
   19 SCMIN = 20.                                                       VORTEX65
```

```
      DO 20 I=1,NSSW                                                    VORTEX66
20    SCWMIN=AMIN1(SCWMIN,TBLSCW(I))                                    VORTEX67
      NSCWMIN=SCWMIN                                                    VORTEX68
      MM=NSCWMIN*NMAX                                                   VORTEX69
      DELTXOC=1./SCWMIN                                                 VORTEX70
      DO 100 LA=1,NSSW                                                  VORTEX71
      CHD(LA)=CCAV(2,LA)*CAVE/BETA                                      VORTEX72
      DELTXX=1./TBLSCW(LA)                                              VORTEX73
      XC=-.75*DELTXX                                                    VORTEX74
      ITBL=TBLSCW(LA)                                                   VORTEX75
      DO 30 LB=1,ITBL                                                   VORTEX76
      XC=XC+DELTXX                                                      VORTEX77
      XXCC(LB)=XC                                                       VORTEX78
      LC=LB+TBLS                                                        VORTEX79
30    ALO(LB)=ALP(LC)                                                   VORTEX80
      XLE=PN(LC)+CHD(LA)*(1.-.75/TBLSCW(LA))                            VORTEX81
      XOC=-.75*DELTXOC                                                  VORTEX82
      KCODE=LB=0                                                        VORTEX83
      DO 90 K=1,NSCWMIN                                                 VORTEX84
      J=K+(LA-1)*NSCWMIN                                                VORTEX85
      XOC=XOC+DELTXOC                                                   VORTEX86
      XCC4(J)=-XOC*CHD(LA)+XLE                                          VORTEX87
      CALL FTLUP (XOC,ALOP(J),+1,ITBL,XXCC,ALO)                         VORTEX88
      AXMN=K*DELTXOC                                                    VORTEX89
      CAT=CATA=0.                                                       VORTEX90
      IF (KCODE.EQ.2) CAT=CCR(LB)-CUT                                   VORTEX91
      IF (KCODE.EQ.2) CATA=CCRR(LB)-CUTA                                VORTEX92
      KCODE=0                                                           VORTEX93
40    LB=LB+1                                                           VORTEX94
      LC=LB+TBLS                                                        VORTEX95
      CCR(LB)=CIR(LC,2)                                                 VORTEX96
      CCRR(LB)=CIR(LC,1)                                                VORTEX97
      AXITBL=LB*DELTXX                                                  VORTEX98
      IF (AXMN-AXITBL) 50,60,70                                         VORTEX99
50    CUT=CCR(LB)*(AXMN-(LB-1)*DELTXX)/DELTXX                           VORTE100
      CUTA=CCRR(LB)*(AXMN-(LB-1)*DELTXX)/DELTXX                         VORTE101
      KCODE=2                                                           VORTE102
      GO TO 80                                                          VORTE103
60    KCODE=1                                                           VORTE104
70    CUT=CCR(LB)                                                       VORTE105
      CUTA=CCRR(LB)                                                     VORTE106
80    CAT=CAT+CUT                                                       VORTE107
      CATA=CATA+CUTA                                                    VORTE108
      IF (KCODE.GE.1) GO TO 85                                          VORTE109
      IF (LB.LT.ITBL) GO TO 40                                          VORTE110
   85 CCC(J)=CAT                                                        VORTE111
      CCH(J)=CATA                                                       VORTE112
   90 CONTINUE                                                          VORTE113
      TBLS=TBLS+TBLSCW(LA)                                              VORTE114
```

```
100   CONTINUE                                          VORTE115
      II=1                                              VORTE116
      IC = 0                                            VORTE117
      IZ = 0                                            VORTE118
      NST = 0                                           VORTE119
      NSTI = 0                                          VORTE120
      DO 150 I=1,IPLAN                                  VORTE121
      IUZ=NSSWSV(I)                                     VORTE122
      IUX=IUZ+1                                         VORTE123
      IF (STLOIND(I) .EQ. 1.) IUX = IUZ                 VORTE124
      IC = IC + MSV(I)                                  VORTE125
      ID=IC+1                                           VORTE126
      IZ = IZ + NSSWSV(I)                               VORTE127
      YCAT=0.                                           VORTE128
      IAMM=NMA(I)                                       VORTE129
C                                                       VORTE130
  105 CONTINUE                                          VORTE131
      DO 140 LA=1,NSCWMIN                               VORTE132
      IF (STLOIND(I) .EC. 1.) GO TO 107                 VORTE133
      YC(1)=-PI/2.                                      VORTE134
      CRI(1)=0.                                         VORTE135
      CRJ(1)=0.                                         VORTE136
  107 DO 120 J = 1,IUZ                                  VORTE137
      L=J+1                                             VORTE138
      IF (STLOIND(I) .EQ. 1.) L = J                     VORTE139
      LU = LA + (J-1 + NST) * NSCWMIN                   VORTE140
      ALLP(J)=ALOP(LU)                                  VORTE141
      XC44(J)=XCC4(LU)                                  VORTE142
      CRI(L)=CCC(LU)                                    VORTE143
      CRJ(L)=CCU(LU)                                    VORTE144
      IF (LA.NE.1) GO TO 120                            VORTE145
      JJ = J + NST                                      VORTE146
      ZZH(J)=ZH(JJ)                                     VORTE147
      SA(J)=SSWWA(JJ)                                   VORTE148
      PPHI(J)=PHI(JJ)                                   VORTE149
      YOO(J)=O(II)                                      VORTE150
      II=II+TPLSCW(JJ)                                  VORTE151
      IE=IUZ-J+1                                        VORTE152
      ITL=TRLSCW(IZ)                                    VORTE153
      ID=ID-ITL                                         VORTE154
      IA=ID+ITL                                         VORTE155
      IF (IA.GT.IC) YCAT=YCAT-S(ID)                     VORTE156
      IF (IA.GT.IC) GO TO 110                           VORTE157
      YCAT=YCAT-S(ID)-S(IA)                             VORTE158
110   IZ=IZ-1                                           VORTE159
      YB(IE)=YCAT                                       VORTE160
120   CONTINUE                                          VORTE161
      DO 130 JP=1,IUZ                                   VORTE162
      JZ=JP+1                                           VORTE163
```

93

```
      IF (FTLFIND(I) .EQ. 1.) J7 = JP                              VORTEF164
      YC(J7) = ASIN(YB(JP)/BOTL(I))                                VORTEF165
  130 CONTINUE                                                     VORTEF166
      YOB=-NMA(I)*2.*SNN-SNN                                       VORTEF167
      DO 140 K=1,IAMM                                              VORTEF168
      KP = LA + (K-1 + NSTT)*NSCWMIN                               VORTEF169
      YOB=YOB+DELTYB                                               VORTEF170
      YOC = ASIN(YOB/BOTL(I))                                      VORTEF171
      CALL FTLUP (YOB,YO(KP),+1,IUZ,YB,YOO)                        VORTEF172
      CALL FTLUP (YOB,ALPPD(KP),+1,IUZ,YB,ALLP)                    VORTEF173
      CALL FTLUP (YOB,SSA(KP),+1,IUZ,YB,SA)                        VORTEF174
      CALL FTLUP (YOB,XC4(KP),+1,IUZ,YB,XC44)                      VORTEF175
      CALL FTLUP (YOB,Z(KP),+1,IUZ,YB,ZZH)                         VORTEF176
      CALL FTLUP (YOB,PHII(KP),+1,IUZ,YB,PPHI)                     VORTEF177
      CALL FTLUP (YOC,GAM(KP),+1,IUX,YC,CRI)                       VORTEF178
      CALL FTLUP (YOC,GAD(KP),+1,IUX,YC,CRJ)                       VORTEF179
      IF (YOB.GT.YB(IUZ)) GAM(KP)=CRI(IUX)                         VORTEF180
      IF (YOB.GT.YB(IUZ)) GAD(KP)=CRJ(IUX)                         VORTEF181
  140 CONTINUE                                                     VORTEF182
      NST = NSSWSV(I) + NST                                        VORTEF183
      NSTT = NMA(I) + NSTT                                         VORTEF184
  150 CONTINUE                                                     VORTEF185
      DO 151 IALPH=1,26                                            VORTEF186
      CDRAG(IALPH)=CLIFT(IALPH)=CPITCH(IALPH)=CSUCT(IALPH)=0.      VORTEF187
      CLAUG(IALPH)=CDAUG(IALPH)=CMAUG(IALPH)=0.                    VORTEF188
  151 CONTINUE                                                     VORTEF189
      CONST=4.*SNN/SREF                                            VORTEF190
C                                                                  VORT1   1
C                                                                  VORT1   2
      DO 1512 J = 1,4                                              VORT1   3
        DO 1510 IALPH = 1,26                                       VORT1   4
          CDS(IALPH,J) = 0.0                                       VORT1   5
          CLF(IALPH,J) = 0.0                                       VORT1   6
          CPH(IALPH,J) = 0.0                                       VORT1   7
          CST(IALPH,J) = 0.0                                       VORT1   8
 1510   CONTINUE                                                   VORT1   9
 1512 CONTINUE                                                     VORT1  10
C                                                                  VORT1  11
C                                                                  VORT1  12
      REWIND 10                                                    VORTEF191
      DO 190 LI=1,NMAX                                             VORTEF192
      IROT = 1                                                     VORTEF193
      KIROT = IPLAN - 1                                            VORTEF194
      DO 1515 ITT = 1,KIROT                                        VORTEF195
        IF(LI .GT. NMASUM(ITT)) IROT = IROT + 1                    VORTEF196
 1515 CONTINUE                                                     VORTEF197
      LA=(LI-1)*NSCWMIN+1                                          VORTEF198
      LB=LI*NSCWMIN                                                VORTEF199
      DO 152 IALPH=1,26                                            VORTEF200
```

```
      CDPAGIT(IALPH)=CLL(IALPH)=CMP(IALPH)=CSS(IALPH)=CLV(IALPH)=0.       VORTE201
  152 CONTINUE                                                             VORTE202
      AAP=ATAN(ALPPD(LA))                                                  VORTE203
      DLYLF=(YC4(LA)+0.25*(XC4(LA)-XC4(LA+1)))*BETA                        VORTE204
      SWLE=ATAN(SSA(LA))                                                   VORTE205
      DO 170 NV=LA,LR                                                      VORTE206
      CPT=COS(ATAN(PHII(NV)))                                              VORTE207
      DO 155 IALPH=1,26                                                    VORTE208
      ALPHA=(IALPH-6)*2.*PI/180.                                           VORTE209
      GAMM=GAM(LA)*SIN(ALPHA)+GAO(LA)                                      VORTE210
      IF(GAMM.EQ.0.) GO TO 153                                             VORTE211
      CUTE(IALPH)=ABS(GAMM)/GAMM                                           VORTE212
      GO TO 154                                                            VORTE213
  153 CUTE(IALPH)=1.                                                       VORTE214
  154 VELIN(IALPH)=0.                                                      VORTE215
  155 CONTINUE                                                             VORTE216
      IF (ICOUNT .GT. 1) GO TO 171                                         VORTE217
      DO 170 NN=1,MM                                                       VORTE218
      XX=XC4(NV)-XC4(NN)                                                   VORTE219
      YY(1)=YQ(NV)-YQ(NN)                                                  VORTE220
      YY(2)=YQ(NV)+YQ(NN)                                                  VORTE221
      ZZ=Z(NV)-Z(NN)                                                       VORTE222
      APHI=ATAN(PHII(NN))                                                  VORTE223
      DO 160 I=1,2                                                         VORTE224
      YYY=YY(I)                                                            VORTE225
      CALL INFSUB (BOT,FV(I),FW(I),FUT)                                    VORTE226
      APHI=-APHI                                                           VORTE227
160   CONTINUE                                                             VORTE228
      DO 165 IALPH=1,26                                                    VORTE229
      ALPHA=(IALPH-6)*2.*PI/180.                                           VORTE230
      GAMM=GAM(NN)*SIN(ALPHA)+GAO(NN)                                      VORTE231
      VELIN(IALPH)=((FW(1)+FW(2))-(FV(1)+FV(2))*PHII(NV))*GAMM/FPI+VELINVORTE232
     1(IALPH)                                                              VORTE233
  165 CONTINUE                                                             VORTE234
170   CONTINUE                                                             VORTE235
      DO 167 IALPH=1,26                                                    VORTE236
      WRITE(20)VELIN(IALPH)                                                VORTE237
  167 CONTINUE                                                             VORTE238
      GO TO 172                                                            VORTE239
  171 DO 156 IALPH=1,26                                                    VORTE240
      READ(20)VELIN(IALPH)                                                 VORTE241
  156 CONTINUE                                                             VORTE242
  172 CONTINUE                                                             VORTE243
      AAPP=ATAN(ALPPD(NV))                                                 VORTE244
      DO 175 IALPH=1,26                                                    VORTE245
      ALPHA=(IALPH-6)*2.*PI/180.                                           VORTE246
      GAMM=GAM(NV)*SIN(ALPHA)+GAO(NV)                                      VORTE247
      CUV=(COS(ALPHA+AAPP)+VELIN(IALPH)*SIN(AAPP))*CPT                     VORTE248
      CUW=(SIN(ALPHA+AAPP)-VELIN(IALPH)*COS(AAPP))*CPT                     VORTE249
```

```
      SOW=SIN(ALPHA+AAPP)
      COW=COS(ALPHA+AAPP)
      COU=TAN(ALPHA+AAP)/COS(SWLE)
      COA=1./(COS(ALPHA+AAP)*COS(SWLE))
      IF(YO(NV).LT.YCHHI(IROT).OR.YO(NV).GT.YCHLO(IROT)) GO TO 173
C
C     LEADING EDGE VORTEX FLOW ANALYSIS EMPLOYED HERE
C
      CN1=CUW*COW*COU*CUTE(IALPH)
      CN2=CUW*COW/COS(SWLE)*CUTE(IALPH)
      CN3=CUW*COW*COS(AAP)*COA*DLXLE*CUTE(IALPH)
      CN4=CUW*COW/(COS(ALPHA+AAP)*COS(SWLE))
      CSS(IALPH)=CSS(IALPH)+GAMM*CN4*2.
      CLV(IALPH)=CLV(IALPH)+GAMM*CN2*2.
      GO TO 174
C
C     ATTACHED FLOW ANALYSIS EMPLOYED HERE
C     ONE HUNDRED PERCENT LEADING EDGE SUCTION
C
  173 IF(ICOUNT.EQ.1) GO TO 177
      CN1=-CUW*COW
      CN2=CUW*COW*TAN(ALPHA+AAP)
      CN3=CUW*COW*SIN(AAP)*DLXLE/COS(ALPHA+AAP)
      GO TO 174
C
C     ATTACHED FLOW ANALYSIS EMPLOYED HERE
C     ZERO PERCENT LEADING EDGE SUCTION
C
  177 CN1=0.
      CN2=0.
      CN3=0.
C
C
  174 CONTINUE
      CDRAGIT(IALPH)=CDRAGIT(IALPH)+GAMM*(CUV*SOW+CN1)*2.
      CLL(IALPH)=CLL(IALPH)+GAMM*(CUV*COW+CN2)*2.
      CMM(IALPH)=CMM(IALPH)+GAMM*(CUV*COS(AAPP)*YC4(NV)*BETA+CN3)*2.
  175 CONTINUE
  180 CONTINUE
      DO 185 IALPH = 1,26
        WRITE(10) CLL(IALPH), CMM(IALPH), CDRAGIT(IALPH)
        IF (ICOUNT .EQ. 3) WRITE(10) CLV(IALPH)
        CDG(IALPH,IROT) = CDG(IALPH,IROT) + CDRAGIT(IALPH) * CONST
        CLF(IALPH,IROT) = CLF(IALPH,IROT) + CLL(IALPH) * CONST
        CPH(IALPH,IROT) = CPH(IALPH,IROT) + CMM(IALPH) * CONST/CREF
        IF (ICOUNT .EQ. 3)
     *    CST(IALPH,IROT) = CST(IALPH,IROT) + CSS(IALPH) * CONST
  185 CONTINUE
  190   CONTINUE
```

```
      END FILE 10                                                       VORTE299
      DO 194 IALPH = 1,26                                               VORTE300
      DO 194 ITT = 1,IPLAN                                              VORTE301
        CDRAG(IALPH) = CDRAG(IALPH) + CDG(IALPH,ITT)                    VORTE302
        CLIFT(IALPH) = CLIFT(IALPH) + CLF(IALPH,ITT)                    VORTE303
        CPITCH(IALPH) = CPITCH(IALPH) + CPH(IALPH,ITT)                  VORTE304
        IF (ICOUNT .EQ. 3)                                              VORTE305
     $    CSUCT(IALPH) = CSUCT(IALPH) + CST(IALPH,ITT)                  VORTE306
  194 CONTINUE                                                          VORTE307
C                                                                       VORTE308
      IF(ICOUNT.EQ.1) END FILE 20                                       VORTE309
      IPUT=0                                                            VORTE310
      REWIND 10                                                         VORTE311
      IF(ICOUNT.NE.3) GO TO 193                                         VORTE312
      CSUCMIN=1.                                                        VORTE313
      DO 192 IJ=1,26                                                    VORTE314
      CSUCMIN=AMIN1(CSUCMIN,CSUCT(IJ))                                  VORTE315
      IF(CSUCMIN.EQ.CSUCT(IJ)) IALPSV=IJ                                VORTE316
  192 CONTINUE                                                          VORTE317
  193 DO 225 IALP=1,26                                                  VORTE318
      ALPHA=(IALP-6)*2.                                                 VORTE319
      WRITE(6,265) ALPHA                                                VORTE320
      IF(ICOUNT.NE.3.OR.DISTALF.EQ.0.) GO TO 191                        VORTE321
      AAL=ALPHA*PI/180.                                                 VORTE322
      JSIGN=1                                                           VORTE323
      IF(IALP.LE.IALPSV) JSIGN=-1                                       VORTE324
      TERM=CSUCT(IALP)*CTILDA/DISTALF*JSIGN                             VORTE325
      CLAUG(IALP)=TERM*COS(AAL)                                         VORTE326
      CDAUG(IALP)=TERM*SIN(AAL)                                         VORTE327
      CMAUG(IALP)=TERM*XTILDA/CREF                                      VORTE328
  191 TPLE=II=0                                                         VORTE329
      LI=0                                                              VORTE330
      IPUT=IPUT+1                                                       VORTE331
      NSTT = 0                                                          VORTE332
      DO 220 I = 1,IPLAN                                                VORTE333
      IAMM=NMA(I)                                                       VORTE334
C                                                                       VORTE335
      DO 200 J=1,IAMM                                                   VORTE336
      JJ = J + NSTT                                                     VORTE337
      LA = 1 + (J - 1 + NSTT) * NSCWMIN                                 VORTE338
      DO 195 IALPH=1,26                                                 VORTE339
      READ(10) CLL(IALPH),CMM(IALPH),CDRAGIT(IALPH)                     VORTE340
      IF(ICOUNT.EQ.3)  READ(10) CLV(IALPH)                              VORTE341
  195 CONTINUE                                                          VORTE342
      CCC(JJ) = CLL(IPUT)/CAVE                                          VORTE343
      CCU(JJ) = CMM(IPUT)/(CAVE*CREF)                                   VORTE344
      XCC4(JJ) = CDRAGIT(IPUT)/CAVE                                     VORTE345
      GAD(JJ) = CLV(IPUT)/CAVE                                          VORTE346
      GAM(J)=CCC(JJ)                                                    VORTE347
```

```
      XC4(J)=CCM(JJ)                                                     VORTE348
      Z(J)=XCC4(JJ)                                                      VORTE349
      SSA(J)=GAM(JJ)                                                     VORTE350
      PHII(J)=YO(LA)                                                     VORTE351
 200  CONTINUE                                                           VORTE352
      IUZ=NSSWSV(I)                                                      VORTE353
      DO 210 LBLAIR=1,IUZ                                                VORTE354
      IF(LBLAIR .EQ. 1)WRITE(6,240)NUMBER(I)                             VORTE355
      LI=LI+1                                                            VORTE356
      LL=1+TBLE                                                          VORTE357
      YBB=Q(LL)                                                          VORTE358
      II=II+1                                                            VORTE359
      TBLE=TBLE+TBLSCW(II)                                               VORTE360
      YOOB=YBB/BOT                                                       VORTE361
      CALL FTLUP (YBB,CLLO,+1,IAMM,PHII,GAM)                             VORTE362
      CALL FTLUP (YBB,CMMO,+1,IAMM,PHII,XC4)                             VORTE363
      CALL FTLUP (YBB,CDRAGO,+1,IAMM,PHII,Z)                             VORTE364
      CALL FTLUP(YBB,CLLV,+1,IAMM,PHII,SSA)                              VORTE365
      WRITE (6,260) LI,YOOB,CLLO,CDRAGO,CMMO,CLLV                        VORTE366
 210  CONTINUE                                                           VORTE367
      NSTT = NMA(I) + NSTT                                               VORTE368
C                                                                        VORTE369
      WRITE(6,271) NUMBER(I), CLF(IALP,I), CDG(IALP,I), CPH(IALP,I)      VORTE370
      IF (ICOUNT .EQ. 3)WRITE(6,285) CST(IALP,I)                         VORTE371
C                                                                        VORTE372
 220  CONTINUE                                                           VORTE373
  221 WRITE(6,270) CLIFT(IALP) ,CDRAG(IALP) ,CPITCH(IALP)                VORTE374
      WRITE(30) CLIFT(IALP),CDRAG(IALP),CPITCH(IALP)                     VORTE375
      REWIND 10                                                          VORTE376
      IF(ICOUNT.EQ.3) WRITE(6,265) CSUCT(IALP)                           VORTE377
  225 CONTINUE                                                           VORTE378
      IF(ICOUNT.NE.3) GO TO 310                                          VORTE379
      DO 300 IALP=1,26                                                   VORTE380
      WRITE(30)CLAUG(IALP),CDAUG(IALP),CMAUG(IALP)                       VORTE381
  300 CONTINUE                                                           VORTE382
  310 ICOUNT=ICOUNT+1                                                    VORTE383
      IF(ICOUNT.EQ.2) GO TO 5                                            VORTE384
      IF (ICOUNT .GT. 3) GO TO 400                                       VORTE385
      DO 320 ITT=1,IPLAN                                                 VORTE386
        YCPLO(ITT) = YINNER(ITT)                                         VORTE387
        YCPHI(ITT) = YOUTER(ITT)                                         VORTE388
  320 CONTINUE                                                           VORTE389
      GO TO 5                                                            VORTE390
C                                                                        VORTE391
C                                                                        VORTE392
  400 CONTINUE                                                           VORTE393
      END FILE 30                                                        VORTE394
      REWIND 30                                                          VORTE395
C                                                                        VORTE396
```

```
C                                                                       VORTE397
C                                                                       VORTE398
  240 FORMAT(/,37X,22HDISTRIBUTIONS FOR THE ,A10,9H PLANFORM ,/)        VORTE399
C                                                                       VORTE400
  250 FORMAT(1H1/////////////////////                                  VORTE401
     1           40X*AERODYNAMIC CHARACTERISTICS FOR CAMBERED AND TWISTEVORTE402
     2D WINGS*/47X*WITH VORTEX LIFT AT VARIOUS ANGLES OF ATTACK*)       VORTE403
  255 FORMAT(//30X*PLANFORM *I2* HAS LEADING EDGE VORTEX FLOW ASSUMED FRVORTE404
     1OM *,F12.5,* TO *,F12.5,/,45X,                                    VORTE405
     2*AND ATTACHED FLOW ELSEWHERE ACROSS THE SPAN*)                    VORTE406
  260 FORMAT(17X,I2,10X,5(F9.5,11X))                                    VORTE407
  265 FORMAT(1H1,////,50X*ANGLE OF ATTACK = *F9.5,1X*DEGREES*//60X*SECTIVORTE408
     1ONAL CHARACTERISTICS*/14X*STATION*,12X,*2Y/B*, 14X9HCL*C/CAVE,    VORTE409
     211X9HCD*C/CAVE,4X21H(CM*C**2)/(CAVE*CREF),4X,17HCL VORT LE*C/CAVE,VORTE410
     3 /90X10HABOUT C.G./)                                              VORTE411
C                                                                       VORTE412
  270 FORMAT(///,50X,21HTOTAL CHARACTERISTICS ,//,20X,                  VORTE413
     $ 4HCL =,F12.5,17X,4HCD =,F12.5,17X,4HCM =,F12.5)                  VORTE414
C                                                                       VORTE415
  271 FORMAT(//,50X,A6,25H PLANFORM CHARACTERISTICS ,//,20X,            VORTE416
     $ 4HCL =,F12.5,17X,4HCD =,F12.5,17X,4HCM =,F12.5)                  VORTE417
C                                                                       VORTE418
  275 FORMAT(///40X*ZERO PERCENT LEADING EDGE SUCTION ASSUMED*)         VORTE419
  280 FORMAT(///36X*ONE HUNDRED PERCENT LEADING EDGE SUCTION ASSUMED*)  VORTE420
C                                                                       VORTE421
  285 FORMAT(//,50X,11HC SUCTION =,F12.5)                               VORTE422
C                                                                       VORTE423
  290 FORMAT(///,10X,30H*** THE VALUE OF -VIC- IS TOO ,                 VORTE424
     $ 39HSMALL. PLEASE CORRECT AND RESUBMIT. *** )                     VORTE425
C                                                                       VORTE426
C                                                                       VORTE427
      RETURN                                                            VORTE428
      END                                                               VORTE429
       SUBROUTINE CNLONG                                                CNLONG 2
C                                                                       CNLONG 3
C               THIS OVERLAY COMPUTES XNUM LOCAL CN VALUES              CNLONG 4
C               AND TOTAL CN FOR EACH PLANFORM.                         CNLONG 5
C                                                                       CNLONG 6
C               ROBERT GRAY  COMPUTER SCIENCES CORP.  1980              CNLONG 7
C                                                                       CNLONG 8
C                                                                       CNLONG 9
       COMMON /ALL/ BOT, BOTSV(4), M, BETA, PTEST, QTEST,               ALL    2
      $    STA(4), TBLSCW(100), YYCP(4),                                ALL    3
      $    Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),        ALL    4
      $    PHI(100), ZH(100), CP(400), STLOIND(4)                       ALL    5
C                                                                       ALL    6
       COMMON /TOTHREE/ CIR(400,2)                                      TOTHRE 2
C                                                                       TOTHRE 3
       COMMON /THREFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD              THREFOR2
C                                                                       THREFOR3
       COMMON /ONETHRE/ TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,   ONETHRE2
      $   ARTRUE, RTCOHT(4), CONFIG(2), NSSWSV(4),                      ONETHRE3
      $   MSV(4), KBOT, PLAN, IPLAN, MACH,                              ONETHRE4
```

```
      $   SSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),       ONETHRE5
      $  CLAMAR(4), CLWIN(4), CLWNG(4), XLOCTN,                           ONETHRE6
      $  YINNER(4), YOUTER(4)                                             ONETHRE7
C                                                                         ONETHRE8
       INTEGER CONFIG                                                     ONETHRE9
C                                                                         ONETHR10
       COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4),  MAINONE2
      $  XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),        MAINONE3
      $  XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),     MAINONE4
      $  ZZ(25,4), ITIPCOD, ICAMTST                                       MAINONE5
C                                                                         MAINONE6
       COMMON/CCRROD/ TSPAN(4), TSPANA, KBIT, CTILDA, XTILDA, DISTALE     CCRROD 2
C                                                                         CCRROD 3
C                                                                         CNLONG16
       DIMENSION DCP(1,25),YL(25),XXL(25),                                CNLONG17
      $ WK1(49),XOVERL(25),CNL(25),YLOBL2(25),IENDSW(2),                  CNLONG18
      $ RPTS(2),CPX(400)                                                  CNLONG19
       DATA XNUM  /20./                                                   CNLONG20
       DATA YNUM  /10./                                                   CNLONG21
       DATA IST   /1/                                                     CNLONG22
       DATA CN /0.0/                                                      CNLONG23
C                                                                         CNLONG24
C                                                                         CNLONG25
       WRITE(6,805)                                                       CNLONG26
C                                                                         CNLONG27
C                                                                         CNLONG28
C                                                                         CNLONG29
C                    FIND X COORDINATES FOR ALREADY COMPUTED              CNLONG30
C                           DELTA CP'S                                    CNLONG31
C                                                                         CNLONG32
       DO 5 I = 1,M                                                       CNLONG33
       CPX(I) = PN(I) * BETA                                              CNLONG34
    5  CONTINUE                                                           CNLONG35
C                                                                         CNLONG36
C                    COMPUTE CN FOR EACH PLANFORM                         CNLONG37
C                                                                         CNLONG38
       DO 1000 IP = 1,IPLAN                                               CNLONG39
C                                                                         CNLONG40
       WRITE(6,806) IP                                                    CNLONG41
       WRITE(6,807)                                                       CNLONG42
C                                                                         CNLONG43
C                    -IST- STARTING POSITION IN ARRAYS                    CNLONG44
C                          CPX,Q, AND CP FOR EACH PLANFORM                CNLONG45
C                    -MAX- NUMBER OF POINTS IN ARRAYS CPX, Q,             CNLONG46
C                          AND CP USED BY EACH PLANFORM                   CNLONG47
C                    -N-   NUMBER OF BREAKPOINTS USED TO DEFINE           CNLONG48
C                          EACH PLANFORM                                  CNLONG49
C                                                                         CNLONG50
       IF (IP .GT. 1) IST = IST + MSV(IP - 1)                             CNLONG51
```

```
      N = AN(IP) + 1                                                    CNLONG52
      MAX = MSV(IP)                                                     CNLONG53
      XFORE = -999999                                                   CNLONG54
      XAFT = 999999                                                     CNLONG55
      YMIN = 999999                                                     CNLONG56
      YROOT = -999999                                                   CNLONG57
C                                                                       CNLONG58
C                  FIND LEADING AND TRAILING POINTS                     CNLONG59
      DO 10 I = 1,N                                                     CNLONG60
        IF (XFORE .LT. XX(I,IP)) XFORE = XX(I,IP)                       CNLONG61
        IF (XAFT  .GT. XX(I,IP)) XAFT = XX(I,IP)                        CNLONG62
        IF (YMIN  .GT. YY(I,IP)) YMIN = YY(I,IP)                        CNLONG63
        IF (YROOT .LT. YY(I,IP)) YROOT = YY(I,IP)                       CNLONG64
   10 CONTINUE                                                          CNLONG65
C                                                                       CNLONG66
C                  FIND TOTAL LENGTH -RL-                               CNLONG67
      RL = ABS(XFORE - XAFT)                                            CNLONG68
C                                                                       CNLONG69
C                  FIND THE X INCREMENT                                 CNLONG70
      XINC = RL/XNUM                                                    CNLONG71
C                  USE MID POINT OF EACH X SEGMENT                      CNLONG72
C                  FOR EACH LOCAL VALUE OF X                            CNLONG73
      XNEXT = XFORE - (XINC / 2.0)                                      CNLONG74
C                                                                       CNLONG75
C                                                                       CNLONG76
C                                                                       CNLONG77
C                   FOR EACH VALUE OF XXL...                            CNLONG78
      NUMX = XNUM                                                       CNLONG79
      N = AN(IP)                                                        CNLONG80
C                                                                       CNLONG81
      DO 100 I = 1,NUMX                                                 CNLONG82
      XXL(I) = XNEXT                                                    CNLONG83
C                   FIND INTERCEPT POINT AT THIS XXL                    CNLONG84
      NPTS = 0                                                          CNLONG85
      DO 30 J = 1,N                                                     CNLONG86
        RXMAX = AMAX1(XX(J,IP),XX(J+1,IP))                              CNLONG87
        RXMIN = AMIN1(XX(J,IP),XX(J+1,IP))                              CNLONG88
        IF (XXL(I) .LE. RXMIN .OR. XXL(I) .GE. RXMAX) GO TO 30          CNLONG89
C                   PERPENDICULAR TO XXL WILL INTERCEPT                 CNLONG90
        NPTS = NPTS + 1                                                 CNLONG91
        IF (NPTS .LE. 2) GO TO 25                                       CNLONG92
        WRITE (6,900) XXL(I)                                            CNLONG93
        GO TO 95                                                        CNLONG94
C                   FIND Y COORDINATE FOR INTERCEPT                     CNLONG95
   25   RYMAX = AMAX1(YY(J,IP),YY(J+1,IP))                              CNLONG96
        RYMIN = AMIN1(YY(J,IP),YY(J+1,IP))                              CNLONG97
C                                                                       CNLONG98
        A = ABS(RXMIN - RXMAX)                                          CNLONG99
        AP = ABS(XXL(I) - RXMAX)                                        CNLON100
```

```
      B = ABS(RYMAX - RYMIN)                                            CNLON101
      BP = (B * AP) / A                                                 CNLON102
C                                                                       CNLON103
      RPTS(NPTS) = RYMAX - BP                                           CNLON104
   30 CONTINUE                                                          CNLON105
C                                                                       CNLON106
      IF(NPTS .EQ. 0) GO TO 90                                          CNLON107
C                                                                       CNLON108
C                   SET BL AND COMPUTE Y COORDINATES                    CNLON109
C                   ON NOTCHED PORTION OF THE WING                      CNLON110
C                   YL GOES FROM THE LEADING EDGE TO THE INNER          CNLON111
C                   EDGE (I.E., FROM YLE TO RYMAX); OTHERWISE,          CNLON112
C                   FROM YLE TO YROOT.                                  CNLON113
      RYMAX = YROOT                                                     CNLON114
      YLE = RPTS(1)                                                     CNLON115
      IF (NPTS .EQ. 1) GO TO 35                                         CNLON116
      RYMIN = AMIN1(RPTS(1),RPTS(2))                                    CNLON117
      RYMAX = AMAX1(RPTS(1),RPTS(2))                                    CNLON118
      YLE = RYMIN                                                       CNLON119
   35 BL = (RYMAX - YLE) * 2                                            CNLON120
      WRITE(6,800)XXL(I),BL                                             CNLON121
      YINC = (BL / 2.) / (YNUM - 1)                                     CNLON122
      NUMY = YNUM                                                       CNLON123
      YNEXT = 0.0                                                       CNLON124
      DO 40 J = 1,NUMY                                                  CNLON125
      YL(J) = YLE + YNEXT                                               CNLON126
      YNEXT = YNEXT + YINC                                              CNLON127
   40 CONTINUE                                                          CNLON128
C                                                                       CNLON129
C                   INTERPOLATE FOR NEW DELTA CP'S                      CNLON130
C                                                                       CNLON131
      CALL INTERP(CPX(IST),Q(IST),CP(IST),MAX,XXL(I),1,YL,NUMY,         CNLON132
     $            DCP,1,IER)                                            CNLON133
C                                                                       CNLON134
      DO 45 J = 1,NUMY                                                  CNLON135
      WRITE(6,801) XXL(I),YL(J),DCP(J)                                  CNLON136
   45 CONTINUE                                                          CNLON137
C                                                                       CNLON138
C                    COMPUTE CNL AT THIS XXL                            CNLON139
C                    = BL/BOT TIMES INTEGRAL FROM 2 * YLE / BL TO       CNLON140
C                      2 * YROOT(OR INNER EDGE OF WING) / BL OF DELTA CP CNLON141
C                      TIMES D(2Y/BL)                                   CNLON142
                                                                        CNLON143
C                    COMPUTE 2Y/BL FOR EACH YL                          CNLON144
      DO 50 J = 1,NUMY                                                  CNLON145
        YLOBL2(J) = (YL(J) / BL) * 2.                                   CNLON146
   50 CONTINUE                                                          CNLON147
C                                                                       CNLON148
C                                                                       CNLON149
```

```
      IENDSW(1) = 1
      IENDSW(2) = 1
      XLOW = YLOBL2(1)
      XUP = YLOBL2(NUMY)
C     IF (NPTS .EQ. 2) XUP = 2 * RYMAX / BL
      SIGMA = 0.0
      IW = 0
      CALL SUTS(NUMY,YLOBL2,DCP,XLOW,XUP,SIGMA,IENDSW,
     $  END,IW,CNL(I),WK1,IER)
C
      CNL(I) = CNL(I) * (BL/BOT)
      WRITE(6,802) CNL(I)
C
       GO TO 95
   90 WRITE(6,901) I,XXL(I)
C
   95 XOVERL(I) = XXL(I) / RL
      XNEXT = XNEXT - XINC
  100 CONTINUE
C
C                    COMPUTE CN =
C                    LENGTH* (B/2)/SREF * INTEGRAL FROM LEADING X/L
C                    TO TRAILING X/L OF CNL TIMES D(X/LENGTH)
       XLOW = XAFT / RL
      XUP = XFORE / RL
      SIGMA = 0.0
      IW = 0
      IENDSW(1) = 1
      IENDSW(2) = 1
C
C                   ROUTINE SUTS REQUIRES THAT VARIABLES
C                   BE ENTERED IN INCREASING ORDER
C
      LIM = NUMX / 2
      DO 110 J = 1,LIM
       K = (NUMX + 1) - J
       RKEEP = XOVERL(J)
       XOVERL(J) = XOVERL(K)
       XOVERL(K) = RKEEP
C
       RKEEP = CNL(J)
       CNL(J) = CNL(K)
       CNL(K) = RKEEP
  110 CONTINUE
C
      CALL SUTS(NUMX,XOVERL,CNL,XLOW,XUP,SIGMA,IENDSW,
     $  IEND,IW,TCN,WK1,IER)
      TCN = TCN * RL * BOT/SREF
      CN = TCN + CN
```

```
      WRITE(6,803) IP,TCN                                               CNLON199
C                                                                       CNLON200
 1000 CONTINUE                                                          CNLON201
C                                                                       CNLON202
      WRITE(6,804) CN                                                   CNLON203
C                                                                       CNLON204
C                                                                       CNLON205
  800 FORMAT(5X,F12.5,42X,F12.5)                                        CNLON206
  801 FORMAT(5X,F12.5,5X,F12.5,5X,F12.5)                                CNLON207
  802 FORMAT (72X,6HCNL = ,F12.5)                                       CNLON208
  803 FORMAT(1H0,17HCN FOR PLANFORM  , I1,3H = ,F12.5)                  CNLON209
  804 FORMAT(1H0,11HTOTAL CN = ,F12.5)                                  CNLON210
  805 FORMAT(1H1,50X,30HLONGITUDINAL LOAD DISTRIBUTION)                 CNLON211
  806 FORMAT(1H0,57X,17HPLANFORM NUMBER  ,I1)                           CNLON212
  807 FORMAT(1H0,10X,1HX,16X,1HY,13X,12HINTERPOLATED,9X,5HBL(X),/,      CNLON213
     $       44X,8HDELTA CP,//)                                         CNLON214
  900  FORMAT (1H ,27HTOO MANY INTERCEPTS AT X = , F12.5,               CNLON215
     $        2X,17HTWO ARE PERMITTED)                                  CNLON216
  901 FORMAT(1H ,27HNO INTERCEPT FOR X STATION ,I2,                     CNLON217
     $  4HX = ,F12.7)                                                   CNLON218
C                                                                       CNLON219
 9000 CONTINUE                                                          CNLON220
      END                                                               CNLON221
      SUBROUTINE INTERP(CPX,Q,CP,MAX,XXL,NX,YL,NUMY,DCP,                INTERP 2
     $                  ND,IER)                                         INTERP 3
C                                                                       INTERP 4
C             THIS SUBROUTINE CALLS ROUTINE -IQHSCV- TO INTERPOLATE     INTERP 5
C             VALUES FOR DELTA CP (DCP).                                INTERP 6
C                                                                       INTERP 7
C                                                                       INTERP 8
C             CPX--  X COORDINATES                                      INTERP 9
C             Q---   Y COORDINATES                                      INTERP10
C             CP--   DELTA CP VALUES AT CPX,Q                           INTERP11
C             MAX--  NUMBER OF POSITIONS TO BE USED IN CPX,Q, AND CP    INTERP12
C             XXL--  ARRAY CONTAINING NEW X CORDINATES                  INTERP13
C             NX--   NUMBER OF ELEMENTS IN XXL                          INTERP14
C             YL--   ARRAY CONTAINING NEW Y COORDINATES                 INTERP15
C             NUMY-  NUMBER OF ELEMENTS IN YL                           INTERP16
C             DCP-   OUTPUT ARRAY DIMENSIONED NX BY NY CONTAINING       INTERP17
C                    INTERPOLATED DELTA CP'S                            INTERP18
C             ND--   ROW DIMENSION OF ARRAY DCP                         INTERP19
C             IWK-   INTEGER WORK ARRAY OF LENGTH                       INTERP20
C                    311 * MAX + NX * NUMY                              INTERP21
C             WK--   REAL WORK ARRAY OF LENGTH 6 * MAX                  INTERP22
C             IER-   ERROR RETURN                                       INTERP23
C                    = 129, MAX IS LESS THAN 4 OR NX OR NUMY            INTERP24
C                           IS LESS THAN 1                              INTERP25
C                    = 130, DATA POINTS ARE COLINEAR                    INTERP26
C                    = 131, SOME DATA POINTS ARE IDENTICAL              INTERP27
```

```
C                                                                       INTERP28
C                                                                       INTERP29
      DIMENSION CPX(1),Q(1),CP(1),XXL(1),YL(1),DCP(1,NUMY)              INTERP30
      DIMENSION IWK(12425),WK(2400)                                     INTERP31
C                                                                       INTERP32
      CALL IQHSCV(CPX,Q,CP,MAX,XXL,NX,YL,NUMY,                          INTERP33
     $            DCP,ND,IWK,WK,IER)                                    INTERP34
C                                                                       INTERP35
C                                                                       INTERP36
      RETURN                                                            INTERP37
      END                                                               INTERP38
      SUBROUTINE IQHSCV (XD,YD,ZD,ND,XI,NXI,YI,NYI,ZI,IZI,IWK,WK,IER)   IQH    2
      INTEGER            ND,NXI,NYI,IZI,IER,IWK(1)                      IQH    3
      REAL               XD(1),YD(1),ZD(1),XI(1),YI(1),ZI(IZI,1),WK(1)  IQH    4
      INTEGER            IL1,IL2,ITI,IXI,IYI,IZ,JIGOMN,JIGOMX,JIG1MN,   IQH    5
     1                   JIG1MX,JIGP,JNGP,JWIGPO,JWIGP,JWIPL,JWIPT,     IQH    6
     2                   JWIWL,JWIWP,JWNGPO,JWNGP,JWWPD,NDPO,NGPO,NGP1, IQH    7
     3                   NL,NNGP,NT,NXIO,NYIO                           IQH    8
      INTEGER            ITPV                                           IQH    9
      REAL               DMMY(27)                                       IQH   10
      COMMON /IBCDPT/    DMMY,ITPV                                      IQH   11
      IER = 0                                                           IQH   12
      NDPO = ND                                                         IQH   13
      NXIO = NXI                                                        IQH   14
      NYIO = NYI                                                        IQH   15
      IF (NDPO.LT.4) GO TO 30                                           IQH   16
      IF (NXIO.LT.1.OR.NYIO.LT.1) GO TO 30                              IQH   17
      IWK(1) = NDPO                                                     IQH   18
      IWK(3) = NXIO                                                     IQH   19
      IWK(4) = NYIO                                                     IQH   20
      JWIPT = 16                                                        IQH   21
      JWIWL = 6*NDPO+1                                                  IQH   22
      JWNGPO = JWIWL-1                                                  IQH   23
      JWIPL = 24*NDPO+1                                                 IQH   24
      JWIWP = 30*NDPO+1                                                 IQH   25
      JWIGPO = 31*NDPO                                                  IQH   26
      JWWPD = 5*NDPO+1                                                  IQH   27
      CALL IQHSG (NDPO,XD,YD,NT,IWK(JWIPT),NL,IWK(JWIPL),IWK(JWIWL),    IQH   28
     1 IWK(JWIWP),WK,IER)                                               IQH   29
      IF (IER.GE.128) GO TO 9000                                        IQH   30
      IWK(5) = NT                                                       IQH   31
      IWK(6) = NL                                                       IQH   32
      IF (NT.EQ.0) GO TO 9005                                           IQH   33
      CALL IQHSH (XD,YD,NT,IWK(JWIPT),NL,IWK(JWIPL),NXIO,NYIO,XI,YI,    IQH   34
     1 IWK(JWNGPO+1),IWK(JWIGPO+1))                                     IQH   35
      CALL IQHSE (NDPO,XD,YD,ZD,NT,IWK(JWIPT),WK,WK(JWWPD))             IQH   36
      ITPV = 0                                                          IQH   37
      JIGOMX = 0                                                        IQH   38
      JIG1MN = NXIO*NYIO+1                                              IQH   39
```

```
      NNGP = NT+2*NL
      DO 25 JNGP=1,NNGP
         ITI = JNGP
         IF (JNGP.LE.NT) GO TO 5
         IL1 = (JNGP-NT+1)/2
         IL2 = (JNGP-NT+2)/2
         IF (IL2.GT.NL) IL2 = 1
         ITI = IL1*(NT+NL)+IL2
    5    JWNGP = JWNGPO+JNGP
         NGPO = IWK(JWNGP)
         IF (NGPO.EQ.0) GO TO 15
         JIGOMN = JIGOMX+1
         JIGOMX = JIGOMX+NGPO
         DO 10 JIGP=JIGOMN,JIGOMX
            JWIGP = JWIGPO+JIGP
            IZ = IWK(JWIGP)
            IYI = (IZ-1)/NXIO+1
            IXI = IZ-NXIO*(IYI-1)
            CALL IQHSF (XD,YD,ZD,NT,IWK(JWIPT),NL,IWK(JWIPL),WK,ITI,
     1      XI(IXI),YI(IYI),ZI(IXI,IYI))
   10    CONTINUE
   15    JWNGP = JWNGPO+2*NNGP+1-JNGP
         NGP1 = IWK(JWNGP)
         IF (NGP1.EQ.0) GO TO 25
         JIG1MX = JIG1MN-1
         JIG1MN = JIG1MN-NGP1
         DO 20 JIGP=JIG1MN,JIG1MX
            JWIGP = JWIGPO+JIGP
            IZ = IWK(JWIGP)
            IYI = (IZ-1)/NXIO+1
            IXI = IZ-NXIO*(IYI-1)
            CALL IQHSF (XD,YD,ZD,NT,IWK(JWIPT),NL,IWK(JWIPL),WK,ITI,
     1      XI(IXI),YI(IYI),ZI(IXI,IYI))
   20    CONTINUE
   25 CONTINUE
      GO TO 9005
   30 IER = 129
 9000 CONTINUE
      CALL UERTST (IER,6HIQHSCV)
 9005 RETURN
      END
      INTEGER FUNCTION IQHSD (X,Y,I1,I2,I3,I4)
      INTEGER            I1,I2,I3,I4
      REAL               X(1),Y(1)
      INTEGER            IDX
      REAL               A1SQ,A2SQ,A3SQ,A4SQ,B1SQ,B2SQ,B3SQ,B4SQ,C1SQ,
     1                   C2SQ,C3SQ,C4SQ,EPSLN,S1SQ,S2SQ,S3SQ,S4SQ,TOL,
     2                   U1,U2,U3,U4,X1,X2,X3,X4,Y1,Y2,Y3,Y4
      EQUIVALENCE        (C2SQ,C1SQ),(A3SQ,B2SQ),(B3SQ,A1SQ),
```

```
     1                        (A4SQ,B1SQ),(B4SQ,A2SQ),(C4SQ,C3SQ)
      DATA                    TOL/16414000000000000000B/
C     TOL = .710543E-14
      EPSLN = TOL*100.0
      X1 = X(I1)
      Y1 = Y(I1)
      X2 = X(I2)
      Y2 = Y(I2)
      X3 = X(I3)
      Y3 = Y(I3)
      X4 = X(I4)
      Y4 = Y(I4)
      IDX = 0
      U3 = (Y2-Y3)*(X1-X3)-(X2-X3)*(Y1-Y3)
      U4 = (Y1-Y4)*(X2-X4)-(X1-X4)*(Y2-Y4)
      IF (U3*U4.LE.0.0) GO TO 5
      U1 = (Y3-Y1)*(X4-X1)-(X3-X1)*(Y4-Y1)
      U2 = (Y4-Y2)*(X3-X2)-(X4-X2)*(Y3-Y2)
      A1SQ = (X1-X3)**2+(Y1-Y3)**2
      B1SQ = (X4-X1)**2+(Y4-Y1)**2
      C1SQ = (X3-X4)**2+(Y3-Y4)**2
      A2SQ = (X2-X4)**2+(Y2-Y4)**2
      B2SQ = (X3-X2)**2+(Y3-Y2)**2
      C3SQ = (X2-X1)**2+(Y2-Y1)**2
      S1SQ = U1*U1/(C1SQ*AMAX1(A1SQ,B1SQ))
      S2SQ = U2*U2/(C2SQ*AMAX1(A2SQ,B2SQ))
      S3SQ = U3*U3/(C3SQ*AMAX1(A3SQ,B3SQ))
      S4SQ = U4*U4/(C4SQ*AMAX1(A4SQ,B4SQ))
      IF ((AMIN1(S3SQ,S4SQ)-AMIN1(S1SQ,S2SQ)).GT.EPSLN) IDX = 1
    5 IQHSD = IDX
      RETURN
      END
      SUBROUTINE IQHSE  (NDP,XD,YD,ZD,NT,IPT,PD,WK)
      INTEGER                 NDP,NT,IPT(1)
      REAL                    XD(1),YD(1),ZD(1),PD(1),WK(1)
      INTEGER                 IDP,IPTI(3),IT,IV,JPDO,JPDMX,JPD,JPTO,JPT,NDPO,
     1                        NTO
      REAL                    DX1,DX2,DY1,DY2,DZ1,DZ2,DZX1,DZX2,DZY1,DZY2,
     1                        EPSLN,TOL,XV(3),VPXX,VPXY,VPX,VPYX,VPYY,VPY,
     2                        VPZMN,VPZ,YV(3),ZV(3),ZXV(3),ZYV(3)
      DATA                    TOL/16414000000000000000B/
C     TOL = .710543E-14
      EPSLN = TOL*100.0
      NDPO = NDP
      NTO = NT
      JPDMX = 5*NDPO
      DO 5 JPD=1,JPDMX
         PD(JPD) = 0.0
    5 CONTINUE
```

```
      DO 10 IDP=1,NDP
         WK(IDP) = 0.0
   10 CONTINUE
      DO 25 IT=1,NTO
         JPTO = 3*(IT-1)
         DO 15 IV=1,3
            JPT = JPTO+IV
            IDP = IPT(JPT)
            IPTI(IV) = IDP
            XV(IV) = XD(IDP)
            YV(IV) = YD(IDP)
            ZV(IV) = ZD(IDP)
   15    CONTINUE
         DX1 = XV(2)-XV(1)
         DY1 = YV(2)-YV(1)
         DZ1 = ZV(2)-ZV(1)
         DX2 = XV(3)-XV(1)
         DY2 = YV(3)-YV(1)
         DZ2 = ZV(3)-ZV(1)
         VPX = DY1*DZ2-DZ1*DY2
         VPY = DZ1*DX2-DX1*DZ2
         VPZ = DX1*DY2-DY1*DX2
         VPZMN = ABS(DX1*DX2+DY1*DY2)*EPSLN
         IF (ABS(VPZ).LE.VPZMN) GO TO 25
         DO 20 IV=1,3
            IDP = IPTI(IV)
            JPDO = 5*(IDP-1)+1
            PD(JPDO) = PD(JPDO)+VPX
            PD(JPDO+1) = PD(JPDO+1)+VPY
            WK(IDP) = WK(IDP)+VPZ
   20    CONTINUE
   25 CONTINUE
      DO 30 IDP=1,NDPO
         JPDO = 5*(IDP-1)+1
         PD(JPDO) = -PD(JPDO)/WK(IDP)
         PD(JPDO+1) = -PD(JPDO+1)/WK(IDP)
   30 CONTINUE
      DO 45 IT=1,NTO
         JPTO = 3*(IT-1)
         DO 35 IV=1,3
            JPT = JPTO+IV
            IDP = IPT(JPT)
            IPTI(IV) = IDP
            XV(IV) = XD(IDP)
            YV(IV) = YD(IDP)
            JPDO = 5*(IDP-1)+1
            ZXV(IV) = PD(JPDO)
            ZYV(IV) = PD(JPDO+1)
   35    CONTINUE
```

```
      DX1 = XV(2)-XV(1)
      DY1 = YV(2)-YV(1)
      DZX1 = ZXV(2)-ZXV(1)
      DZY1 = ZYV(2)-ZYV(1)
      DX2 = XV(3)-XV(1)
      DY2 = YV(3)-YV(1)
      DZX2 = ZXV(3)-ZXV(1)
      DZY2 = ZYV(3)-ZYV(1)
      VPXX = DY1*DZX2-DZX1*DY2
      VPXY = DZX1*DX2-DX1*DZX2
      VPYX = DY1*DZY2-DZY1*DY2
      VPYY = DZY1*DX2-DX1*DZY2
      VPZ = DX1*DY2-DY1*DX2
      VPZMN = ABS(DX1*DX2+DY1*DY2)*EPSLN
      IF (ABS(VPZ).LE.VPZMN) GO TO 45
      DO 40 IV=1,3
         IDP = IPTI(IV)
         JPDO = 5*(IDP-1)+3
         PD(JPDO) = PD(JPDO)+VPXX
         PD(JPDO+1) = PD(JPDO+1)+VPXY+VPYX
         PD(JPDO+2) = PD(JPDO+2)+VPYY
   40 CONTINUE
   45 CONTINUE
      DO 50 IDP=1,NDPO
         JPDO = 5*(IDP-1)+3
         PD(JPDO) = -PD(JPDO)/WK(IDP)
         PD(JPDO+1) = -PD(JPDO+1)/(2.0*WK(IDP))
         PD(JPDO+2) = -PD(JPDO+2)/WK(IDP)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE IQHSF  (XD,YD,ZD,NT,IPT,NL,IPL,PDD,ITI,XII,YII,ZII)
      INTEGER           NT,NL,ITI,IPT(1),IPL(1)
      REAL              XD(1),YD(1),ZD(1),PDD(1),XII,YII,ZII
      INTEGER           IDP,IL1,IL2,ITO,I,JIPL,JIPT,JPDD,JPD,KPD,NTL
      INTEGER           ITPV
      REAL              XO,YO,AP,BP,CP,DP,P00,P10,P20,P30,P40,P50,P5,
     1                  P01,P11,P21,P31,P41,P02,P12,P22,P32,P03,P13,
     2                  P23,P04,P14,P05
      REAL              AA,AB,ACT2,AC,ADBC,AD,A,BB,BC,BDT2,B,CC,CD,
     1                  CSUV,C,DD,DLT,DX,DY,D,G1,G2,H1,H2,H3,LU,
     2                  LV,X(3),PO,P1,P2,P3,P4,PD(15),THSV,THUS,THUV,
     3                  THXU,U,V,Y(3),ZO,ZUU(3),ZUV(3),ZU(3),ZVV(3),
     4                  ZV(3),Z(3)
      COMMON /IBCDPT/   XO,YO,AP,BP,CP,DP,P00,P10,P20,P30,P40,P50,
     1                  P01,P11,P21,P31,P41,P02,P12,P22,P32,P03,P13,
     2                  P23,P04,P14,P05,ITPV
      EQUIVALENCE       (P5,P50)
      ITO = ITI
```

```
      NTL = NT+NL
      IF (ITO.LE.NTL) GO TO 5
      IL1 = ITO/NTL
      IL2 = ITO-IL1*NTL
      IF (IL1.EQ.IL2) GO TO 30
      GO TO 55
    5 IF (ITO.EQ.ITPV) GO TO 25
      JIPT = 3*(ITO-1)
      JPD = 0
      DO 15 I=1,3
         JIPT = JIPT+1
         IDP = IPT(JIPT)
         X(I) = XD(IDP)
         Y(I) = YD(IDP)
         Z(I) = ZD(IDP)
         JPDD = 5*(IDP-1)
         DO 10 KPD=1,5
            JPD = JPD+1
            JPDD = JPDD+1
            PD(JPD) = PDD(JPDD)
   10    CONTINUE
   15 CONTINUE
      XO = X(1)
      YO = Y(1)
      A = X(2)-XO
      B = X(3)-XO
      C = Y(2)-YO
      D = Y(3)-YO
      AD = A*D
      BC = B*C
      DLT = AD-BC
      AP = D/DLT
      BP = -B/DLT
      CP = -C/DLT
      DP = A/DLT
      AA = A*A
      ACT2 = 2.0*A*C
      CC = C*C
      AB = A*B
      ADBC = AD+BC
      CD = C*D
      BB = B*B
      BDT2 = 2.0*B*D
      DD = D*D
      DO 20 I=1,3
         JPD = 5*I
         ZU(I) = A*PD(JPD-4)+C*PD(JPD-3)
         ZV(I) = B*PD(JPD-4)+D*PD(JPD-3)
         ZUU(I) = AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)
```

```
      ZUV(I) = AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)
      ZVV(I) = BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)
   20 CONTINUE
      P00 = Z(1)
      P10 = ZU(1)
      P01 = ZV(1)
      P20 = 0.5*ZUU(1)
      P11 = ZUV(1)
      P02 = 0.5*ZVV(1)
      H1 = Z(2)-P00-P10-P20
      H2 = ZU(2)-P10-ZUU(1)
      H3 = ZUU(2)-ZUU(1)
      P30 = 10.0*H1-4.0*H2+0.5*H3
      P40 = -15.0*H1+7.0*H2-H3
      P50 = 6.0*H1-3.0*H2+0.5*H3
      H1 = Z(3)-P00-P01-P02
      H2 = ZV(3)-P01-ZVV(1)
      H3 = ZVV(3)-ZVV(1)
      P03 = 10.0*H1-4.0*H2+0.5*H3
      P04 = -15.0*H1+7.0*H2-H3
      P05 = 6.0*H1-3.0*H2+0.5*H3
      LU = SQRT(AA+CC)
      LV = SQRT(BB+DD)
      THXU = ATAN2(C,A)
      THUV = ATAN2(D,B)-THXU
      CSUV = COS(THUV)
      P41 = 5.0*LV*CSUV/LU*P50
      P14 = 5.0*LU*CSUV/LV*P05
      H1 = ZV(2)-P01-P11-P41
      H2 = ZUV(2)-P11-4.0*P41
      P21 = 3.0*H1-H2
      P31 = -2.0*H1+H2
      H1 = ZU(3)-P10-P11-P14
      H2 = ZUV(3)-P11-4.0*P14
      P12 = 3.0*H1-H2
      P13 = -2.0*H1+H2
      THUS = ATAN2(D-C,B-A)-THXU
      THSV = THUV-THUS
      AA = SIN(THSV)/LU
      BB = -COS(THSV)/LU
      CC = SIN(THUS)/LV
      DD = COS(THUS)/LV
      AC = AA*CC
      AD = AA*DD
      BC = BB*CC
      G1 = AA*AC*(3.0*BC+2.0*AD)
      G2 = CC*AC*(3.0*AD+2.0*BC)
      H1 = -AA*AA*AA*(5.0*AA*BB*P50+(4.0*BC+AD)*P41)-CC*CC*CC*(5.0*CC
     1*DD*P05+(4.0*AD+BC)*P14)
```

```
      H2 = 0.5*ZVV(2)-P02-P12
      H3 = 0.5*ZUU(3)-P20-P21
      P22 = (G1*H2+G2*H3-H1)/(G1+G2)
      P32 = H2-P22
      P23 = H3-P22
      ITPV = ITO
   25 DX = XII-XO
      DY = YII-YO
      U = AP*DX+BP*DY
      V = CP*DX+DP*DY
      PO = P00+V*(P01+V*(P02+V*(P03+V*(P04+V*P05))))
      P1 = P10+V*(P11+V*(P12+V*(P13+V*P14)))
      P2 = P20+V*(P21+V*(P22+V*P23))
      P3 = P30+V*(P31+V*P32)
      P4 = P40+V*P41
      ZII = P0+U*(P1+U*(P2+U*(P3+U*(P4+U*P5))))
      RETURN
   30 IF (ITO.EQ.ITPV) GO TO 50
      JIPL = 3*(IL1-1)
      JPD = 0
      DO 40 I=1,2
         JIPL = JIPL+1
         IDP = IPL(JIPL)
         X(I) = XD(IDP)
         Y(I) = YD(IDP)
         Z(I) = ZD(IDP)
         JPDD = 5*(IDP-1)
         DO 35 KPD=1,5
            JPD = JPD+1
            JPDD = JPDD+1
            PD(JPD) = PDD(JPDD)
   35    CONTINUE
   40 CONTINUE
      XO = X(1)
      YO = Y(1)
      A = Y(2)-Y(1)
      B = X(2)-X(1)
      C = -B
      D = A
      AD = A*D
      BC = B*C
      DLT = AD-BC
      AP = D/DLT
      BP = -B/DLT
      CP = -BP
      DP = AP
      AA = A*A
      ACT2 = 2.0*A*C
      CC = C*C
```

```
      AB = A*B
      ADBC = AD+BC
      CD = C*D
      BB = B*B
      BDT2 = 2.0*B*D
      DD = D*D
      DO 45 I=1,2
         JPD = 5*I
         ZU(I) = A*PD(JPD-4)+C*PD(JPD-3)
         ZV(I) = B*PD(JPD-4)+D*PD(JPD-3)
         ZUU(I) = AA*PD(JPD-2)+ACT2*PD(JPD-1)+CC*PD(JPD)
         ZUV(I) = AB*PD(JPD-2)+ADBC*PD(JPD-1)+CD*PD(JPD)
         ZVV(I) = BB*PD(JPD-2)+BDT2*PD(JPD-1)+DD*PD(JPD)
   45 CONTINUE
      P00 = Z(1)
      P10 = ZU(1)
      P01 = ZV(1)
      P20 = 0.5*ZUU(1)
      P11 = ZUV(1)
      P02 = 0.5*ZVV(1)
      H1 = Z(2)-P00-P01-P02
      H2 = ZV(2)-P01-ZVV(1)
      H3 = ZVV(2)-ZVV(1)
      P03 = 10.0*H1-4.0*H2+0.5*H3
      P04 = -15.0*H1+7.0*H2-H3
      P05 = 6.0*H1-3.0*H2+0.5*H3
      H1 = ZU(2)-P10-P11
      H2 = ZUV(2)-P11
      P12 = 3.0*H1-H2
      P13 = -2.0*H1+H2
      P21 = 0.0
      P23 = -ZUU(2)+ZUU(1)
      P22 = -1.5*P23
      ITPV = ITO
   50 DX = XII-X0
      DY = YII-Y0
      U = AP*DX+BP*DY
      V = CP*DX+DP*DY
      P0 = P00+V*(P01+V*(P02+V*(P03+V*(P04+V*P05))))
      P1 = P10+V*(P11+V*(P12+V*P13))
      P2 = P20+V*(P21+V*(P22+V*P23))
      ZII = P0+U*(P1+U*P2)
      RETURN
   55 IF (ITO.EQ.ITPV) GO TO 65
      JIPL = 3*IL2-2
      IDP = IPL(JIPL)
      X0 = XD(IDP)
      Y0 = YD(IDP)
      Z0 = ZD(IDP)
```

```
      JPDD = 5*(IDP-1)
      DO 60 KPD=1,5
         JPDD = JPDD+1
         PD(KPD) = PDD(JPDD)
   60 CONTINUE
      P00 = Z0
      P10 = PD(1)
      P01 = PD(2)
      P20 = 0.5*PD(3)
      P11 = PD(4)
      P02 = 0.5*PD(5)
      ITPV = IT0
   65 U = XII-X0
      V = YII-Y0
      P0 = P00+V*(P01+V*P02)
      P1 = P10+V*P11
      ZII = P0+U*(P1+U*P20)
      RETURN
      END
      SUBROUTINE IQHSG  (NDP,XD,YD,NT,IPT,NL,IPL,IWL,IWP,WK,IER)
      INTEGER            NDP,NT,NL,IER,IPT(1),IPL(1),IWL(1),IWP(1)
      REAL               XD(1),YD(1),WK(1)
      INTEGER            ILF,ILIV,ILT3,ILVS,IL,IP1P1,IP1,IP2,IP3,IPL1,
     1                   IPL2,IPLJ1,IPLJ2,IPMN1,IPMN2,IPT1,IPT2,IPT3,
     2                   IPTI1,IPTI2,IPTI,IP,IREP,IT1T3,IT2T3,ITF(2),
     3                   ITS,ITT3R,ITT3,IT,IXVSPV,IXVS,JL1,JL2,JLT3,JP1,
     4                   JP2,JPC,JPMN,JPMX,JP,JWL1MN,JWL1,JWL,NDP0,
     5                   NDPM1,NL0,NLFC,NLFT2,NLF,NLNT3,NLN,NLSHT3,NLSH,
     6                   NLT3,NREP,NT0,NTF,IQHSD,NTT3P3,NTT3
      REAL               DSQI,DSQMN,EPSLN,SP,TOL,VP,X1,X2,X3,XDMP,Y1,Y2,
     1                   Y3,YDMP
      REAL               DSQF,SPDT,VPDT,U1,U2,U3,V1,V2,V3
      DATA               NREP/100/
      DATA               TOL/16414000000000000000B/
C     TOL = .710543E-14
      DSQF(U1,V1,U2,V2) = (U2-U1)**2+(V2-V1)**2
      SPDT(U1,V1,U2,V2,U3,V3) = (U2-U1)*(U3-U1)+(V2-V1)*(V3-V1)
      VPDT(U1,V1,U2,V2,U3,V3) = (V3-V1)*(U2-U1)-(U3-U1)*(V2-V1)
      EPSLN = TOL*100.0
      NDP0 = NDP
      NDPM1 = NDP0-1
      DSQMN = DSQF(XD(1),YD(1),XD(2),YD(2))
      IPMN1 = 1
      IPMN2 = 2
      DO 10 IP1=1,NDPM1
         X1 = XD(IP1)
         Y1 = YD(IP1)
         IP1P1 = IP1+1
         DO 5 IP2=IP1P1,NDP0
```

```
      DSQI = DSQF(X1,Y1,XD(IP2),YD(IP2))
      IF (DSQI.EQ.0.0) GO TO 160
      IF (DSQI.GE.DSQMN) GO TO 5
      DSQMN = DSQI
      IPMN1 = IP1
      IPMN2 = IP2
    5   CONTINUE
   10 CONTINUE
      XDMP = (XD(IPMN1)+XD(IPMN2))/2.0
      YDMP = (YD(IPMN1)+YD(IPMN2))/2.0
      JP1 = 2
      DO 15 IP1=1,NDPO
        IF (IP1.EQ.IPMN1.OR.IP1.EQ.IPMN2) GO TO 15
        JP1 = JP1+1
        IWP(JP1) = IP1
        WK(JP1) = DSQF(XDMP,YDMP,XD(IP1),YD(IP1))
   15 CONTINUE
      DO 25 JP1=3,NDPM1
        DSQMN = WK(JP1)
        JPMN = JP1
        DO 20 JP2=JP1,NDPO
          IF (WK(JP2).GE.DSQMN) GO TO 20
          DSQMN = WK(JP2)
          JPMN = JP2
   20   CONTINUE
        ITS = IWP(JP1)
        IWP(JP1) = IWP(JPMN)
        IWP(JPMN) = ITS
        WK(JPMN) = WK(JP1)
   25 CONTINUE
      X1 = XD(IPMN1)
      Y1 = YD(IPMN1)
      X2 = XD(IPMN2)
      Y2 = YD(IPMN2)
      DO 30 JP=3,NDPO
        IP = IWP(JP)
        SP = SPDT(XD(IP),YD(IP),X1,Y1,X2,Y2)
        VP = VPDT(XD(IP),YD(IP),X1,Y1,X2,Y2)
        IF (ABS(VP).GT.(ABS(SP)*EPSLN)) GO TO 35
   30 CONTINUE
      GO TO 165
   35 IF (JP.EQ.3) GO TO 45
      JPMX = JP
      DO 40 JPC=4,JPMX
        JP = JPMX+4-JPC
        IWP(JP) = IWP(JP-1)
   40 CONTINUE
      IWP(3) = IP
   45 IP1 = IPMN1
```

```
      IP2 = IPMN2
      IP3 = IWP(3)
      IF (VPDT(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3)).GE.0.0)
     1GO TO 50
      IP1 = IPMN2
      IP2 = IPMN1
   50 NTO = 1
      NTT3 = 3
      IPT(1) = IP1
      IPT(2) = IP2
      IPT(3) = IP3
      NLO = 3
      NLT3 = 9
      IPL(1) = IP1
      IPL(2) = IP2
      IPL(3) = 1
      IPL(4) = IP2
      IPL(5) = IP3
      IPL(6) = 1
      IPL(7) = IP3
      IPL(8) = IP1
      IPL(9) = 1
      DO 150 JP1=4,NDPO
         IP1 = IWP(JP1)
         X1 = XD(IP1)
         Y1 = YD(IP1)
         DO 65 IL=1,NLO
            IP2 = IPL(3*IL-2)
            IP3 = IPL(3*IL-1)
            X2 = XD(IP2)
            Y2 = YD(IP2)
            X3 = XD(IP3)
            Y3 = YD(IP3)
            SP = SPDT(X1,Y1,X2,Y2,X3,Y3)
            VP = VPDT(X1,Y1,X2,Y2,X3,Y3)
            IF (IL.NE.1) GO TO 55
            IXVS = 0
            IF (VP.LE.(ABS(SP)*(-EPSLN))) IXVS = 1
            ILIV = 1
            ILVS = 1
            GO TO 65
   55       IXVSPV = IXVS
            IF (VP.GT.(ABS(SP)*(-EPSLN))) GO TO 60
            IXVS = 1
            IF (IXVSPV.EQ.1) GO TO 65
            ILVS = IL
            IF (ILIV.NE.1) GO TO 70
            GO TO 65
   60       IXVS = 0
```

```
      IF (IXVSPV.EQ.0) GO TO 65                                         IQH  579
      ILIV = IL                                                         IQH  580
      IF (ILVS.NE.1) GO TO 70                                           IQH  581
   65 CONTINUE                                                          IQH  582
      IF (ILIV.EQ.1.AND.ILVS.EQ.1) ILVS = NLO                           IQH  583
   70 IF (ILVS.LT.ILIV) ILVS = ILVS+NLO                                 IQH  584
      IF (ILIV.EQ.1) GO TO 85                                           IQH  585
      NLSH = ILIV-1                                                     IQH  586
      NLSHT3 = NLSH*3                                                   IQH  587
      DO 75 JL1=1,NLSHT3                                                IQH  588
        JL2 = JL1+NLT3                                                  IQH  589
        IPL(JL2) = IPL(JL1)                                             IQH  590
   75 CONTINUE                                                          IQH  591
      DO 80 JL1=1,NLT3                                                  IQH  592
        JL2 = JL1+NLSHT3                                                IQH  593
        IPL(JL1) = IPL(JL2)                                             IQH  594
   80 CONTINUE                                                          IQH  595
      ILVS = ILVS-NLSH                                                  IQH  596
   85 JWL = 0                                                           IQH  597
      DO 105 IL=ILVS,NLO                                                IQH  598
        ILT3 = IL*3                                                     IQH  599
        IPL1 = IPL(ILT3-2)                                              IQH  600
        IPL2 = IPL(ILT3-1)                                              IQH  601
        IT = IPL(ILT3)                                                  IQH  602
        NTO = NTO+1                                                     IQH  603
        NTT3 = NTT3+3                                                   IQH  604
        IPT(NTT3-2) = IPL2                                              IQH  605
        IPT(NTT3-1) = IPL1                                              IQH  606
        IPT(NTT3) = IP1                                                 IQH  607
        IF (IL.NE.ILVS) GO TO 90                                        IQH  608
        IPL(ILT3-1) = IP1                                               IQH  609
        IPL(ILT3) = NTO                                                 IQH  610
   90   IF (IL.NE.NLO) GO TO 95                                         IQH  611
        NLN = ILVS+1                                                    IQH  612
        NLNT3 = NLN*3                                                   IQH  613
        IPL(NLNT3-2) = IP1                                              IQH  614
        IPL(NLNT3-1) = IPL(1)                                           IQH  615
        IPL(NLNT3) = NTO                                                IQH  616
   95   ITT3 = IT*3                                                     IQH  617
        IPTI = IPT(ITT3-2)                                              IQH  618
        IF (IPTI.NE.IPL1.AND.IPTI.NE.IPL2) GO TO 100                    IQH  619
        IPTI = IPT(ITT3-1)                                              IQH  620
        IF (IPTI.NE.IPL1.AND.IPTI.NE.IPL2) GO TO 100                    IQH  621
        IPTI = IPT(ITT3)                                                IQH  622
  100   IF (IQHSD(XD,YD,IP1,IPTI,IPL1,IPL2).EQ.0) GO TO 105             IQH  623
        IPT(ITT3-2) = IPTI                                              IQH  624
        IPT(ITT3-1) = IPL1                                              IQH  625
        IPT(ITT3) = IP1                                                 IQH  626
        IPT(NTT3-1) = IPTI                                              IQH  627
```

```
      IF (IL.EQ.ILVS) IPL(ILT3) = IT
      IF (IL.EQ.NLO.AND.IPL(3).EQ.IT) IPL(3) = NTO
      JWL = JWL+4
      IWL(JWL-3) = IPL1
      IWL(JWL-2) = IPTI
      IWL(JWL-1) = IPTI
      IWL(JWL) = IPL2
  105 CONTINUE
      NLO = NLN
      NLT3 = NLNT3
      NLF = JWL/2
      IF (NLF.EQ.0) GO TO 150
      NTT3P3 = NTT3+3
      DO 145 IREP=1,NREP
         DO 135 ILF=1,NLF
            IPL1 = IWL(2*ILF-1)
            IPL2 = IWL(2*ILF)
            NTF = 0
            DO 110 ITT3R=3,NTT3,3
               ITT3 = NTT3P3-ITT3R
               IPT1 = IPT(ITT3-2)
               IPT2 = IPT(ITT3-1)
               IPT3 = IPT(ITT3)
               IF (IPL1.NE.IPT1.AND.IPL1.NE.IPT2.AND.IPL1.NE.IPT3) GO
     1          TO 110
               IF (IPL2.NE.IPT1.AND.IPL2.NE.IPT2.AND.IPL2.NE.IPT3) GO
     1          TO 110
               NTF = NTF+1
               ITF(NTF) = ITT3/3
               IF (NTF.EQ.2) GO TO 115
  110       CONTINUE
            IF (NTF.LT.2) GO TO 135
  115       IT1T3 = ITF(1)*3
            IPTI1 = IPT(IT1T3-2)
            IF (IPTI1.NE.IPL1.AND.IPTI1.NE.IPL2) GO TO 120
            IPTI1 = IPT(IT1T3-1)
            IF (IPTI1.NE.IPL1.AND.IPTI1.NE.IPL2) GO TO 120
            IPTI1 = IPT(IT1T3)
  120       IT2T3 = ITF(2)*3
            IPTI2 = IPT(IT2T3-2)
            IF (IPTI2.NE.IPL1.AND.IPTI2.NE.IPL2) GO TO 125
            IPTI2 = IPT(IT2T3-1)
            IF (IPTI2.NE.IPL1.AND.IPTI2.NE.IPL2) GO TO 125
            IPTI2 = IPT(IT2T3)
  125       IF (IQHSD(XD,YD,IPTI1,IPTI2,IPL1,IPL2).EQ.0) GO TO 135
            IPT(IT1T3-2) = IPTI1
            IPT(IT1T3-1) = IPTI2
            IPT(IT1T3) = IPL1
            IPT(IT2T3-2) = IPTI2
```

```
            IPT(IT2T3-1) = IPTI1
            IPT(IT2T3) = IPL2
            JWL = JWL+8
            IWL(JWL-7) = IPL1
            IWL(JWL-6) = IPTI1
            IWL(JWL-5) = IPTI1
            IWL(JWL-4) = IPL2
            IWL(JWL-3) = IPL2
            IWL(JWL-2) = IPTI2
            IWL(JWL-1) = IPTI2
            IWL(JWL) = IPL1
            DO 130 JLT3=3,NLT3,3
               IPLJ1 = IPL(JLT3-2)
               IPLJ2 = IPL(JLT3-1)
               IF ((IPLJ1.EQ.IPL1.AND.IPLJ2.EQ.IPTI2).OR.(IPLJ2.EQ.IP
     1         L1.AND.IPLJ1.EQ.IPTI2)) IPL(JLT3)
     2         = ITF(1)
               IF ((IPLJ1.EQ.IPL2.AND.IPLJ2.EQ.IPTI1).OR.(IPLJ2.EQ.IP
     1         L2.AND.IPLJ1.EQ.IPTI1)) IPL(JLT3)
     2         = ITF(2)
  130       CONTINUE
  135    CONTINUE
         NLFC = NLF
         NLF = JWL/2
         IF (NLF.EQ.NLFC) GO TO 150
         JWL1MN = 2*NLFC+1
         NLFT2 = NLF*2
         DO 140 JWL1=JWL1MN,NLFT2
            JWL = JWL1+1-JWL1MN
            IWL(JWL) = IWL(JWL1)
  140    CONTINUE
         NLF = JWL/2
  145 CONTINUE
  150 CONTINUE
      DO 155 ITT3=3,NTT3,3
         IP1 = IPT(ITT3-2)
         IP2 = IPT(ITT3-1)
         IP3 = IPT(ITT3)
         IF (VPDT(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3)).GE.0.
     1   0) GO TO 155
         IPT(ITT3-2) = IP2
         IPT(ITT3-1) = IP1
  155 CONTINUE
      NT = NTO
      NL = NLO
      RETURN
  160 IER = 131
      RETURN
  165 IER = 130
```

```
      RETURN                                                            IQH 726
      END                                                               IQH 727
      SUBROUTINE IQHSH  (XD,YD,NT,IPT,NL,IPL,NXI,NYI,XI,YI,NGP,IGP)     IQH 728
      INTEGER           NT,NL,NXI,NYI,IPT(1),IPL(1),NGP(1),IGP(1)       IQH 729
      REAL              XD(1),YD(1),XI(1),YI(1)                         IQH 730
      INTEGER           ILOT3,ILO,ILP1T3,ILP1,INSD,IP1,IP2,IP3,ITOT3,   IQH 731
     1                  ITO,IXIMN,IXIMX,IXI,IYI,IZI,JIGPO,JIGP1I,JIGP1,IQH 732
     2                  JNGPO,JNGP1,L,NGPO,NGP1,NLO,NTO,NXIO,NXINYI,    IQH 733
     3                  NYIO                                            IQH 734
      REAL              X1,X2,X3,XII,XIMN,XIMX,XMN,XMX,Y1,Y2,Y3,YII,    IQH 735
     1                  YIMN,YIMX,YMN,YMX                               IQH 736
      REAL              SPDT,VPDT,U1,U2,U3,V1,V2,V3                     IQH 737
      SPDT(U1,V1,U2,V2,U3,V3) = (U1-U2)*(U3-U2)+(V1-V2)*(V3-V2)         IQH 738
      VPDT(U1,V1,U2,V2,U3,V3) = (U1-U3)*(V2-V3)-(V1-V3)*(U2-U3)         IQH 739
      NTO = NT                                                          IQH 740
      NLO = NL                                                          IQH 741
      NXIO = NXI                                                        IQH 742
      NYIO = NYI                                                        IQH 743
      NXINYI = NXIO*NYIO                                                IQH 744
      XIMN = AMIN1(XI(1),XI(NXIO))                                      IQH 745
      XIMX = AMAX1(XI(1),XI(NXIO))                                      IQH 746
      YIMN = AMIN1(YI(1),YI(NYIO))                                      IQH 747
      YIMX = AMAX1(YI(1),YI(NYIO))                                      IQH 748
      JNGPO = 0                                                         IQH 749
      JNGP1 = 2*(NTO+2*NLO)+1                                           IQH 750
      JIGPO = 0                                                         IQH 751
      JIGP1 = NXINYI+1                                                  IQH 752
      DO 80 ITO=1,NTO                                                   IQH 753
         NGPO = 0                                                       IQH 754
         NGP1 = 0                                                       IQH 755
         ITOT3 = ITO*3                                                  IQH 756
         IP1 = IPT(ITOT3-2)                                             IQH 757
         IP2 = IPT(ITOT3-1)                                             IQH 758
         IP3 = IPT(ITOT3)                                               IQH 759
         X1 = XD(IP1)                                                   IQH 760
         Y1 = YD(IP1)                                                   IQH 761
         X2 = XD(IP2)                                                   IQH 762
         Y2 = YD(IP2)                                                   IQH 763
         X3 = XD(IP3)                                                   IQH 764
         Y3 = YD(IP3)                                                   IQH 765
         XMN = AMIN1(X1,X2,X3)                                          IQH 766
         XMX = AMAX1(X1,X2,X3)                                          IQH 767
         YMN = AMIN1(Y1,Y2,Y3)                                          IQH 768
         YMX = AMAX1(Y1,Y2,Y3)                                          IQH 769
         INSD = 0                                                       IQH 770
         DO 10 IXI=1,NXIO                                               IQH 771
            IF (XI(IXI).GE.XMN.AND.XI(IXI).LE.XMX) GO TO 5              IQH 772
            IF (INSD.EQ.0) GO TO 10                                     IQH 773
            IXIMX = IXI-1                                               IQH 774
```

```
      GO TO 15
    5 IF (INSD.EQ.1) GO TO 10
      INSD = 1
      IXIMN = IXI
   10 CONTINUE
      IF (INSD.EQ.0) GO TO 75
      IXIMX = NXIO
   15 DO 70 IYI=1,NYIO
        YII = YI(IYI)
        IF (YII.LT.YMN.OR.YII.GT.YMX) GO TO 70
        DO 65 IXI=IXIMN,IXIMX
          XII = XI(IXI)
          L = 0
          IF (VPDT(X1,Y1,X2,Y2,XII,YII)) 65,20,25
   20     L = 1
   25     IF (VPDT(X2,Y2,X3,Y3,XII,YII)) 65,30,35
   30     L = 1
   35     IF (VPDT(X3,Y3,X1,Y1,XII,YII)) 65,40,45
   40     L = 1
   45     IZI = NXIO*(IYI-1)+IXI
          IF (L.EQ.1) GO TO 50
          NGPO = NGPO+1
          JIGPO = JIGPO+1
          IGP(JIGPO) = IZI
          GO TO 65
   50     IF (JIGP1.GT.NXINYI) GO TO 60
          DO 55 JIGP1I=JIGP1,NXINYI
            IF (IZI.EQ.IGP(JIGP1I)) GO TO 65
   55     CONTINUE
   60     NGP1 = NGP1+1
          JIGP1 = JIGP1-1
          IGP(JIGP1) = IZI
   65   CONTINUE
   70 CONTINUE
   75 JNGPO = JNGPO+1
      NGP(JNGPO) = NGPO
      JNGP1 = JNGP1-1
      NGP(JNGP1) = NGP1
   80 CONTINUE
      DO 225 ILO=1,NLO
        NGPO = 0
        NGP1 = 0
        ILOT3 = ILO*3
        IP1 = IPL(ILOT3-2)
        IP2 = IPL(ILOT3-1)
        X1 = XD(IP1)
        Y1 = YD(IP1)
        X2 = XD(IP2)
        Y2 = YD(IP2)
```

```
      XMN = XIMN
      XMX = XIMX
      YMN = YIMN
      YMX = YIMX
      IF (Y2.GE.Y1) XMN = AMIN1(X1,X2)
      IF (Y2.LE.Y1) XMX = AMAX1(X1,X2)
      IF (X2.LE.X1) YMN = AMIN1(Y1,Y2)
      IF (X2.GE.X1) YMX = AMAX1(Y1,Y2)
      INSD = 0
      DO 90 IXI=1,NXIO
         IF (XI(IXI).GE.XMN.AND.XI(IXI).LE.XMX) GO TO 85
         IF (INSD.EQ.0) GO TO 90
         IXIMX = IXI-1
         GO TO 95
   85    IF (INSD.EQ.1) GO TO 90
         INSD = 1
         IXIMN = IXI
   90 CONTINUE
      IF (INSD.EQ.0) GO TO 155
      IXIMX = NXIO
   95 DO 150 IYI=1,NYIO
         YII = YI(IYI)
         IF (YII.LT.YMN.OR.YII.GT.YMX) GO TO 150
         DO 145 IXI=IXIMN,IXIMX
            XII = XI(IXI)
            L = 0
            IF (VPDT(X1,Y1,X2,Y2,XII,YII)) 105,100,145
  100       L = 1
  105       IF (SPDT(X2,Y2,X1,Y1,XII,YII)) 145,110,115
  110       L = 1
  115       IF (SPDT(X1,Y1,X2,Y2,XII,YII)) 145,120,125
  120       L = 1
  125       IZI = NXIO*(IYI-1)+IXI
            IF (L.EQ.1) GO TO 130
            NGPO = NGPO+1
            JIGPO = JIGPO+1
            IGP(JIGPO) = IZI
            GO TO 145
  130       IF (JIGP1.GT.NXINYI) GO TO 140
            DO 135 JIGP1I=JIGP1,NXINYI
               IF (IZI.EQ.IGP(JIGP1I)) GO TO 145
  135       CONTINUE
  140       NGP1 = NGP1+1
            JIGP1 = JIGP1-1
            IGP(JIGP1) = IZI
  145    CONTINUE
  150 CONTINUE
  155 JNGPO = JNGPO+1
      NGP(JNGPO) = NGPO
```

```
      JNGP1 = JNGP1-1
      NGP(JNGP1) = NGP1
      NGPO = 0
      NGP1 = 0
      ILP1 = MOD(ILO,NLO)+1
      ILP1T3 = ILP1*3
      IP3 = IPL(ILP1T3-1)
      X3 = XD(IP3)
      Y3 = YD(IP3)
      XMN = XIMN
      XMX = XIMX
      YMN = YIMN
      YMX = YIMX
      IF (Y3.GE.Y2.AND.Y2.GE.Y1) XMN = X2
      IF (Y3.LE.Y2.AND.Y2.LF.Y1) XMX = X2
      IF (X3.LE.X2.AND.X2.LE.X1) YMN = Y2
      IF (X3.GE.X2.AND.X2.GE.X1) YMX = Y2
      INSD = 0
      DO 165 IXI=1,NXIO
        IF (XI(IXI).GE.XMN.AND.XI(IXI).LE.XMX) GO TO 160
        IF (INSD.EQ.0) GO TO 165
        IXIMX = IXI-1
        GO TO 170
  160   IF (INSD.EQ.1) GO TO 165
        INSD = 1
        IXIMN = IXI
  165 CONTINUE
      IF (INSD.EQ.0) GO TO 220
      IXIMX = NXIO
  170 DO 215 IYI=1,NYIO
        YII = YI(IYI)
        IF (YII.LT.YMN.OR.YII.GT.YMX) GO TO 215
        DO 210 IXI=IXIMN,IXIMX
          XII = XI(IXI)
          L = 0
          IF (SPDT(X1,Y1,X2,Y2,XII,YII)) 180,175,210
  175     L = 1
  180     IF (SPDT(X3,Y3,X2,Y2,XII,YII)) 190,185,210
  185     L = 1
  190     IZI = NXIO*(IYI-1)+IXI
          IF (L.EQ.1) GO TO 195
          NGPO = NGPO+1
          JIGPO = JIGPO+1
          IGP(JIGPO) = IZI
          GO TO 210
  195     IF (JIGP1.GT.NXINYI) GO TO 205
          DO 200 JIGP1I=JIGP1,NXINYI
            IF (IZI.EQ.IGP(JIGP1I)) GO TO 210
  200     CONTINUE
```

```
205          NGP1 = NGP1+1                                              IQH  922
             JIGP1 = JIGP1-1                                            IQH  923
             IGP(JIGP1) = IZI                                           IQH  924
210        CONTINUE                                                     IQH  925
215     CONTINUE                                                        IQH  926
220     JNGPO = JNGPO+1                                                 IQH  927
        NGP(JNGPO) = NGPO                                               IQH  928
        JNGP1 = JNGP1-1                                                 IQH  929
        NGP(JNGP1) = NGP1                                               IQH  930
225 CONTINUE                                                            IQH  931
    RETURN                                                              IQH  932
    END                                                                 IQH  933
    SUBROUTINE UERTST (IER,NAME)                                        IQH  934
    INTEGER            IER,NAME                                         IQH  935
    INTEGER            NAMSET,NAMEQ                                     IQH  936
    DATA               NAMSET/6HUERSET/                                 IQH  937
    DATA               NAMEQ/6H      /                                  IQH  938
    DATA               LEVEL/4/,IEQDF/0/,IEQ/1H=/                       IQH  939
    IF (IER.GT.999) GO TO 25                                            IQH  940
    IF (IER.LT.-32) GO TO 55                                            IQH  941
    IF (IER.LE.128) GO TO 5                                             IQH  942
    IF (LEVEL.LT.1) GO TO 30                                            IQH  943
    CALL UGETIO(1,NIN,IOUNIT)                                           IQH  944
    IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAME                 IQH  945
    IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAME                           IQH  946
    GO TO 30                                                            IQH  947
  5 IF (IER.LE.64) GO TO 10                                             IQH  948
    IF (LEVEL.LT.2) GO TO 30                                            IQH  949
    CALL UGETIO(1,NIN,IOUNIT)                                           IQH  950
    IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAME                 IQH  951
    IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAME                           IQH  952
    GO TO 30                                                            IQH  953
 10 IF (IER.LE.32) GO TO 15                                             IQH  954
    IF (LEVEL.LT.3) GO TO 30                                            IQH  955
    CALL UGETIO(1,NIN,IOUNIT)                                           IQH  956
    IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAME                 IQH  957
    IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAME                           IQH  958
    GO TO 30                                                            IQH  959
 15 CONTINUE                                                            IQH  960
    IF (NAME.NE.NAMSET) GO TO 25                                        IQH  961
    LEVOLD = LEVEL                                                      IQH  962
    LEVEL = IER                                                         IQH  963
    IER = LEVOLD                                                        IQH  964
    IF (LEVEL.LT.0) LEVEL = 4                                           IQH  965
    IF (LEVEL.GT.4) LEVEL = 4                                           IQH  966
    GO TO 30                                                            IQH  967
 25 CONTINUE                                                            IQH  968
    IF (LEVEL.LT.4) GO TO 30                                            IQH  969
    CALL UGETIO(1,NIN,IOUNIT)                                           IQH  970
```

```
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAME               IQH  971
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAME                         IQH  972
   30 IEQDF = 0                                                         IQH  973
      RETURN                                                            IQH  974
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,                   IQH  975
     & 15H) FROM ROUTINE ,A6,A1,A6)                                     IQH  976
   40 FORMAT(36H *** WARNING WITH FIX ERROR  (IER = ,I3,                IQH  977
     $ 15H) FROM ROUTINE ,A6,A1,A6)                                     IQH  978
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,                    IQH  979
     $ 15H) FROM ROUTINE ,A6,A1,A6)                                     IQH  980
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,                   IQH  981
     $ 15H) FROM ROUTINE, A6,A1,A6)                                     IQH  982
   55 IEQDF = 1                                                         IQH  983
      NAMEQ = NAME                                                      IQH  984
   65 RETURN                                                            IQH  985
      END                                                               IQH  986
      SUBROUTINE UGETIO(IOPT,NIN,NOUT)                                  IQH  987
      INTEGER            IOPT,NIN,NOUT                                  IQH  988
      INTEGER            NIND,NOUTD                                     IQH  989
      DATA               NIND/5LINPUT/,NOUTD/6LOUTPUT/                  IQH  990
      IF (IOPT.EQ.3) GO TO 10                                           IQH  991
      IF (IOPT.EQ.2) GO TO 5                                            IQH  992
      IF (IOPT.NE.1) GO TO 9005                                         IQH  993
      NIN = NIND                                                        IQH  994
      NOUT = NOUTD                                                      IQH  995
      GO TO 9005                                                        IQH  996
    5 NIND = NIN                                                        IQH  997
      GO TO 9005                                                        IQH  998
   10 NOUTD = NOUT                                                      IQH  999
 9005 RETURN                                                            IQH 1000
      END                                                               IQH 1001
      SUBROUTINE SUTS(N,X,Y,XL,XU,SIGMA,IENDSW,END,IW,PROXIN,WK,IERR)   SUTS    2
C     D1.11                                                             SUTS    3
C                                                                       SUTS    4
C***********************************************************************SUTS    5
C*                                                                     *SUTS    6
C*                                                                     *SUTS    7
C*   PUPPOSE             - TO COMPUTE THE INTEGRAL OF A CUBIC SPLINE   *SUTS    8
C*                         INTERPOLATION UNDER TENSION TO A SET OF     *SUTS    9
C*                         DISCRETE NODES OF INDEPENDENT AND DEPENDENT *SUTS   10
C*                         REAL VARIABLES.                             *SUTS   11
C*                                                                     *SUTS   12
C*   USE                 - CALL SUTS(N,X,Y,XL,XU,SIGMA,IENDSW,END,IW,  *SUTS   13
C*                                   PROXIN,WK,IERR)                   *SUTS   14
C*                                                                     *SUTS   15
C*   PARAMETERS   N      - AN INPUT INTEGER SPECIFYING THE NUMBER OF   *SUTS   16
C*                         OF NODES OR VALUES FOR BOTH THE INPUT       *SUTS   17
C*                         INDEPENDENT AND DEPENDENT VARIABLES.  N  > 1.*SUTS   18
C*               X       - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *SUTS   19
```

```
C*                  N  CONTAINING THE INDEPENDENT VARIABLES OR   *SUTS  20
C*                  X-COORDINATES OF THE NODES.  X  MUST BE A    *SUTS  21
C*                  STRICTLY INCREASING ARRAY.                   *SUTS  22
C*        Y       - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *SUTS  23
C*                  N  CONTAINING THE DEPENDENT VARIABLES OR     *SUTS  24
C*                  Y-COORDINATES OF THE NODES.  Y  MUST BE A    *SUTS  25
C*                  STRICTLY INCREASING ARRAY.                   *SUTS  26
C*        SIGMA   - AN INPUT REAL NUMBER CONTAINING THE TENSION  *SUTS  27
C*        XL      - AN INPUT REAL NUMBER SPECIFYING THE LOWER    *SUTS  28
C*                  LIMIT OF INTEGRATION.                        *SUTS  29
C*        XU      - AN INPUT REAL NUMBER SPECIFYING THE UPPER    *SUTS  30
C*                  LIMIT OF INTEGRATION.                        *SUTS  31
C*                  FACTOR.  SIGMA  SHOULD BE NON-NEGATIVE.      *SUTS  32
C*        IENDSW  - A ONE-DIMENSIONAL INPUT INTEGER ARRAY OF     *SUTS  33
C*                  LENGTH 2 SPECIFYING CONDITIONS ON THE CURVE  *SUTS  34
C*                  AT THE TWO ENDPOINTS.  IENDSW(1)  REFERS TO  *SUTS  35
C*                  (X(1),Y(1))  AND  IENDSW(2)  REFERS TO       *SUTS  36
C*                  (X(N),Y(N)) .                                *SUTS  37
C*                  IENDSW(I) = 0 MEANS THAT THE USER WILL INPUT *SUTS  38
C*                                A VALUE FOR Y' AT  THE         *SUTS  39
C*                                APPROPRIATE ENDPOINT IN        *SUTS  40
C*                                END(I) .                       *SUTS  41
C*                  IENDSW(I) = 1 MEANS THAT Y' WILL BE ESTIMATED*SUTS  42
C*                                INTERNALLY FOR THE APPROPRIATE *SUTS  43
C*                                ENDPOINT.                      *SUTS  44
C*                  IENDSW(I) = 2 MEANS THAT Y" = ZERO (I.E., THE*SUTS  45
C*                                NATURAL END-CONDITION HOLDS) AT*SUTS  46
C*                                THE APPROPRIATE ENDPOINT.      *SUTS  47
C*        END     - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *SUTS  48
C*                  2 CONTAINING DESIRED VALUES FOR Y' AT        *SUTS  49
C*                  (X(1),Y(1))  IN  END(1)  AND AT  (X(N),Y(N)) *SUTS  50
C*                  IN  END(2) .  IF  IENDSW(I)  .NE. 0, THEN    *SUTS  51
C*                  END(I)  NEED NOT BE DEFINED.  IF BOTH        *SUTS  52
C*                  IENDSW(1)  AND  IENDSW(2)  ARE NON-ZERO, THEN*SUTS  53
C*                  END  MAY BE A DUMMY PARAMETER.               *SUTS  54
C*        IW      - AN INPUT/OUTPUT INTEGER.                     *SUTS  55
C*                  INPUT  - IF THE USER IS INPUTTING  X ,  Y ,  *SUTS  56
C*                           N ,  SIGMA ,  IENDSW , AND  END  FOR*SUTS  57
C*                           THE FIRST TIME (AS A PACKAGE), THEN *SUTS  58
C*                           IW  SHOULD BE SET TO 0.  OTHERWISE, *SUTS  59
C*                           IW  SHOULD BE SET TO 1.             *SUTS  60
C*                  OUTPUT - IW  IS AUTOMATICALLY SET TO 1.      *SUTS  61
C*        PROXIN  - AN OUTPUT REAL NUMBER SPECIFYING THE VALUE OF*SUTS  62
C*                  THE INTEGRAL.                                *SUTS  63
C*        WK      - A ONE-DIMENSIONAL INPUT/OUTPUT/WORK REAL     *SUTS  64
C*                  ARRAY OF LENGTH  2N  - 1.                    *SUTS  65
C*                  INPUT  - IF  IW  = 1, THEN THE FIRST  N      *SUTS  66
C*                           PLACES  OF  WK  MUST BE OUTPUT FROM *SUTS  67
C*                           A PREVIOUS CALL TO  STIUNI (E1.8),  *SUTS  68
```

```
C*                            SUTDER (D4.3), OR  SUTS (D1.11).    *
C*                   OUTPUT - THE FIRST  N  PLACES OF  WK  CONTAIN*
C*                            DERIVATIVE INFORMATION NECESSARY ON *
C*                            SUCCEEDING CALLS TO  STIUNI ,       *
C*                            SUTDER , OR  SUTS  USING THE SAME   *
C*                            INPUT DATA.                         *
C*                   WORK   - THE REST OF  WK  IS WORK STORAGE.   *
C*             IERR   - AN OUTPUT INTEGER ERROR CODE.             *
C*                      = 0  NORMAL RETURN.                       *
C*                      = 1  N  < 2.                              *
C*                      = 2  X  IS NOT AN INCREASING ARRAY.       *
C*                                                                *
C*  PRECISION         - SINGLE                                    *
C*                                                                *
C*  REQUIRED ROUTINES - CEEZ,CURVI,CURV1N,CURV12,INTRVL,SNHCSH,TERMS.*
C*                                                                *
C*  DATE RELEASED     - MARCH 1, 1979.                            *
C*                                                                *
C*  LANGUAGE          - FORTRAN                                   *
C*                                                                *
C*  LATEST REVISION   - NONE                                      *
C*                                                                *
C*                                                                *
C******************************************************************
C
C           FORMAL PARAMETERS
C
      INTEGER IENDSW(2),IERR,IW,N
C
      REAL END(2),PROXIN,SIGMA,WK(1),X(1),XL,XU,Y(1)
C
C           INTERNAL VARIABLES
C
      REAL CURVI,DELX,SIGMAP,ZERO
C
C           CONSTRUCT THE CONSTANT  ZERO
C
      DATA ZERO/0.0E0/
      DATA IUNISW /0/
      ISLPSW=(IENDSW(2)+2*IENDSW(1))
      IUNISW=1
C
C           INITIALIZE  IERR  AND CHECK ERROR RETURNS
C
      IERR = 1
      IF (N .LT. 2) GO TO 9000
      IERR = 2
      DELX = X(N) - X(1)
      IF (DELX .LE. ZERO) GO TO 9000
```

```
      IERR = 0                                                          SUTS 118
C                                                                       SUTS 119
C              DENORMALIZE THE TENSION FACTOR  SIGMA                    SUTS 120
C                                                                       SUTS 121
      SIGMAP = ABS(SIGMA) * (N - 1)/DELX                                SUTS 122
C                                                                       SUTS 123
C              CALL  CURV1  IF NECESSARY                                SUTS 124
C                                                                       SUTS 125
      IF (IW .EQ. 1) GO TO 10                                           SUTS 126
C                                                                       SUTS 127
         IF (N .GT. 2) CALL CURV1N(N,X,Y,IENDSW,END,SIGMAP,WK,WK(N+1),  SUTS 128
     $                              IERR)                               SUTS 129
         IF (N .EQ. 2) CALL CURV12(DELX,Y,IENDSW,END,SIGMAP,WK)         SUTS 130
         IF (IERR .NE. 0) GO TO 9000                                    SUTS 131
C                                                                       SUTS 132
C              SET  IW  = 1 FOR OUTPUT                                  SUTS 133
C                                                                       SUTS 134
            IW = 1                                                      SUTS 135
C                                                                       SUTS 136
C              COMPUTE THE INTEGRAL  PROXIN                             SUTS 137
C                                                                       SUTS 138
   10 PROXIN = CURVI(XL,XU,N,X,Y,WK,SIGMAP)                             SUTS 139
C                                                                       SUTS 140
 9000 RETURN                                                            SUTS 141
      END                                                               SUTS 142
      SUBROUTINE CURV1N(N,X,Y,IENDSW,END,SIGMAP,YP,TEMP,IERR)           SUTS 143
C                                                                       SUTS 144
C***********************************************************************SUTS 145
C*                                                                     *SUTS 146
C*                                                                     *SUTS 147
C*  PURPOSE               - COMPUTE SECOND DERIVATIVES NECESSARY TO BE *SUTS 148
C*                          ABLE TO INTERPOLATE POINTS ON A SPLINE     *SUTS 149
C*                          FUNCTION UNDER TENSION PASSING THROUGH AT  *SUTS 150
C*                          LEAST THREE NODES.  SEE  STIUNI (E1.X).    *SUTS 151
C*                                                                     *SUTS 152
C*  USE                   - CALL CURV1N(N,X,Y,IENDSW,END,SIGMAP,YP,TEMP, *SUTS 153
C*                                  IERR)                              *SUTS 154
C*                                                                     *SUTS 155
C*  PARAMETERS    N       - AN INPUT INTEGER SPECIFYING THE NUMBER OF  *SUTS 156
C*                          NODES.  N  MUST BE AT LEAST 3.             *SUTS 157
C*                X       - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 158
C*                          N  CONTAINING THE X-COORDINATES OF THE NODES.*SUTS 159
C*                Y       - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 160
C*                          N  CONTAINING THE Y-COORDINATES OF THE NODES.*SUTS 161
C*                IENDSW  - AN INPUT ONE-DIMENSIONAL INTEGER ARRAY OF  *SUTS 162
C*                          LENGTH 2 SPECIFYING THE OPTIONS CHOSEN AT THE*SUTS 163
C*                          TWO ENDPOINTS OF THE SPLINE CURVE.  IENDSW(1)*SUTS 164
C*                          REFERS TO  (X(1),Y(1))  AND IENDSW(2)      *SUTS 165
C*                          REFERS TO  (X(N),Y(N)) .                   *SUTS 166
```

```
C*                       IENDSW(I) = 0  THE USER HAS INPUT VALUES FOR *SUTS 167
C*                                      Y' AT THE APPROPRIATE ENDPOINT*SUTS 168
C*                                      IN  END(I) .                  *SUTS 169
C*                                 = 1  Y' WILL BE ESTIMATED          *SUTS 170
C*                                      INTERNALLY AT THE APPROPRIATE *SUTS 171
C*                                      ENDPOINT.                     *SUTS 172
C*                                 = 2  THE NATURAL CONDITION (Y" = 0)*SUTS 173
C*                                      WILL BE IMPOSED ON THE        *SUTS 174
C*                                      APPROPRIATE ENDPOINT.         *SUTS 175
C*            END    - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 176
C*                     2.  END(1)  CONTAINS THE DERIVATIVE OF THE   *SUTS 177
C*                     SPLINE FUNCTION AT  (X(1),Y(1)) .  END(2)    *SUTS 178
C*                     CONTAINS THE DERIVATIVE OF THE SPLINE        *SUTS 179
C*                     FUNCTION AT  (X(N),Y(N)) .  IF  IENDSW(I)  IS*SUTS 180
C*                     NON-ZERO, THEN  END(I)  NEED NOT BE DEFINED. *SUTS 181
C*                     IF BOTH  IENDSW(1)  AND  IENDSW(2)  ARE NON- *SUTS 182
C*                     ZERO, THEN  END  MAY BE A DUMMY PARAMETER.   *SUTS 183
C*            SIGMAP - AN INPUT REAL NUMBER SPECIFYING THE          *SUTS 184
C*                     DENORMALIZED TENSION FACTOR.                 *SUTS 185
C*            YP     - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF      *SUTS 186
C*                     LENGTH  N  CONTAINING SECOND DERIVATIVE      *SUTS 187
C*                     INFORMATION NECESSARY TO INTERPOLATE THE     *SUTS 188
C*                     SPLINE FUNCTION.                             *SUTS 189
C*            TEMP   - A WORK ONE-DIMENSIONAL REAL ARRAY OF LENGTH  *SUTS 190
C*                     N - 1.                                       *SUTS 191
C*            IERR   - AN OUTPUT INTEGER SPECIFYING THE ERROR       *SUTS 192
C*                     RETURN CODE.                                 *SUTS 193
C*                     = 0  NORMAL RETURN.                          *SUTS 194
C*                     = 2  TWO CONSECUTIVE  X-VALUES  ARE THE SAME.*SUTS 195
C*                                                                  *SUTS 196
C*  PRECISION          - SINGLE.                                    *SUTS 197
C*                                                                  *SUTS 198
C*  REQUIRED ROUTINES  - CEEZ,SNHCSH,TERMS.                         *SUTS 199
C*                                                                  *SUTS 200
C*  DATE RELEASED      - MARCH 1, 1979.                             *SUTS 201
C*                                                                  *SUTS 202
C*  LANGUAGE           - FORTRAN.                                   *SUTS 203
C*                                                                  *SUTS 204
C*  SOURCE             - A. K. CLINE AND R. J. RENKA                *SUTS 205
C*                       UNIVERSITY OF TEXAS AT AUSTIN              *SUTS 206
C*                       NON-NATURAL OPTIONS ONLY                   *SUTS 207
C*                                                                  *SUTS 208
C*  LATEST REVISION    - NONE.                                      *SUTS 209
C*                                                                  *SUTS 210
C*                                                                  *SUTS 211
C*******************************************************************SUTS 212
C                                                                    SUTS 213
C           FORMAL PARAMETERS                                        SUTS 214
C                                                                    SUTS 215
```

```
      INTEGER IENDSW(2),IERR,N
C
      REAL END(2),SIGMAP,TEMP(1),X(N),Y(N),YP(N)
C
C             INTERNAL VARIABLES
C
      INTEGER I,IBAK,NM1,NP1
C
      REAL C1,C2,C3,DELXN,DELXNM,DELX1,DELX2,DIAG,DIAG1,DIAG2,DX1,DX2
      REAL SDIAG1,SDIAG2,SLPN,SLP1,ZERO
C
C             DEFINE THE CONSTANT  ZERO
C
      DATA ZERO/0.0E0/
C
C             INITIALIZE VARIABLES
C
      NM1 = N - 1
      NP1 = N + 1
      DELX1 = X(2) - X(1)
      IERR = 2
C
C             APPROXIMATE THE LEFT END SLOPE IF NEEDED
C
      IF (DELX1 .LE. ZERO) GO TO 9000
      IF (IENDSW(1) .EQ. 0) SLP1 = END(1)
      IF (IENDSW(1) .NE. 1) GO TO 10
         DELX2 = X(3) - X(1)
         IF (DELX2 .LE. DELX1) GO TO 9000
         CALL CEEZ (DELX1,DELX2,SIGMAP,C1,C2,C3,N)
         SLP1 = C1*Y(1) + C2*Y(2) + C3*Y(3)
C
C             APPROXIMATE THE RIGHT END SLOPE IF NEEDED
C
   10 IF (IENDSW(2) .EQ. 0) SLPN = END(2)
      IF (IENDSW(2) .NE. 1) GO TO 20
         DELXN = X(N) - X(NM1)
         DELXNM = X(N) - X(N-2)
         IF (DELXN .LE. ZERO .OR. DELXNM .LE. DELXN) GO TO 9000
         CALL CEEZ (-DELXN,-DELXNM,SIGMAP,C1,C2,C3,N)
         SLPN = C1*Y(N) + C2*Y(NM1) + C3*Y(N-2)
C
C             SET UP RIGHT HAND SIDE AND TRIDIAGONAL SYSTEM FOR YP AND
C             PERFORM FORWARD ELIMINATION
C
   20 DX1 = (Y(2) - Y(1)) / DELX1
      CALL TERMS (DIAG1,SDIAG1,SIGMAP,DELX1)
C
C             TEST WHETHER THE LEFT ENDPOINT IS NATURAL
```

```
C
      IF (IENDSW(1) .EQ. 2) GO TO 30
C
C             THIS CODE IS FOR A NON-NATURAL LEFT ENDPOINT
C
         YP(1) = (DX1 - SLP1) / DIAG1
         TEMP(1) = SDIAG1/DIAG1
         GO TO 40
C
C             THIS CODE IS FOR A NATURAL LEFT ENDPOINT
C
   30 TEMP(1) = ZERO
      YP(1) = ZERO
C
C             BEGIN THE ELIMINATION LOOP
C
   40 DO 50 I=2,NM1
         IM1 = I - 1
         DELX2 = X(I+1) - X(I)
         IF (DELX2 .LE. ZERO) GO TO 9000
         DX2 = (Y(I+1) - Y(I)) / DELX2
         CALL TERMS (DIAG2,SDIAG2,SIGMAP,DELX2)
         DIAG = DIAG1 + DIAG2 - SDIAG1*TEMP(IM1)
         YP(I) = (DX2 - DX1 - SDIAG1*YP(IM1)) / DIAG
         TEMP(I) = SDIAG2/DIAG
         DX1 = DX2
         DIAG1 = DIAG2
         SDIAG1 = SDIAG2
   50 CONTINUE
C
C             TEST WHETHER THE RIGHT ENDPOINT IS NATURAL
C
      IF (IENDSW(2) .EQ. 2) GO TO 60
C
C             THIS CODE IS FOR A NON-NATURAL RIGHT ENDPOINT
C
      DIAG = DIAG1 - SDIAG1*TEMP(NM1)
      YP(N) = (SLPN - DX1 - SDIAG1*YP(NM1)) / DIAG
      GO TO 70
C
C             THIS CODE IS FOR A NATURAL RIGHT ENDPOINT
C
   60 YP(N) = ZERO
C
C             PERFORM BACK SUBSTITUTION
C
   70 DO 80 I = 2,N
         IBAK = NP1 - I
         YP(IBAK) = YP(IBAK) - TEMP(IBAK)*YP(IBAK+1)
```

```
   80 CONTINUE                                                          SUTS 314
      IERR = 0                                                          SUTS 315
C                                                                       SUTS 316
 9000 RETURN                                                            SUTS 317
C                                                                       SUTS 318
      END                                                               SUTS 319
      SUBROUTINE CURV12(DX,Y,IENDSW,END,SIGMAP,YP)                      SUTS 320
C                                                                       SUTS 321
C***********************************************************************SUTS 322
C*                                                                     *SUTS 323
C*                                                                     *SUTS 324
C*  PURPOSE                 - COMPUTE SECOND DERIVATIVES NECESSARY TO BE *SUTS 325
C*                            ABLE TO INTERPOLATE POINTS ON A SPLINE    *SUTS 326
C*                            FUNCTION UNDER TENSION PASSING THROUGH    *SUTS 327
C*                            EXACTLY TWO NODES.  SEE  STIUNI (E1.X).   *SUTS 328
C*                                                                     *SUTS 329
C*  USE                     - CALL CURV12(DX,Y,IENDSW,END,SIGMAP,YP    ) *SUTS 330
C*                                                                     *SUTS 331
C*  PARAMETERS    DX        - AN INPUT REAL NUMBER =  X(2) - X(1)  WHERE *SUTS 332
C*                            THE  X'S  ARE THE X-COORDINATES OF THE NODES.*SUTS 333
C*                Y         - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 334
C*                            2 CONTAINING THE Y-COORDINATES OF THE NODES. *SUTS 335
C*                IENDSW    - AN INPUT ONE-DIMENSIONAL INTEGER ARRAY OF  *SUTS 336
C*                            LENGTH 2 SPECIFYING THE OPTIONS CHOSEN AT THE*SUTS 337
C*                            TWO ENDPOINTS OF THE SPLINE CURVE.        *SUTS 338
C*                            IENDSW(I) = 0  THE USER HAS INPUT VALUES FOR *SUTS 339
C*                                           (X(I),Y'(I))  IN  END(I) .   *SUTS 340
C*                                      = 1  (X(I),Y'(I))  WILL BE       *SUTS 341
C*                                           ESTIMATED INTERNALLY.       *SUTS 342
C*                                      = 2  THE NATURAL CONDITION       *SUTS 343
C*                                           ( (X(I),Y"(I)) = 0 ) WILL BE *SUTS 344
C*                                           IMPOSED.                    *SUTS 345
C*                END       - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 346
C*                            2.  END(I)  CONTAINS THE DERIVATIVE OF THE  *SUTS 347
C*                            SPLINE FUNCTION AT  (X(I),Y(I)) .  IF      *SUTS 348
C*                            IENDSW(I)  IS NON-ZERO, THEN  END(I)  NEED  *SUTS 349
C*                            NOT BE DEFINED.  IF BOTH  IENDSW(1)  AND    *SUTS 350
C*                            IENDSW(2)  ARE NON-ZERO. THEN  END  MAY BE A *SUTS 351
C*                            DUMMY PARAMETER.                           *SUTS 352
C*                SIGMAP    - AN INPUT REAL NUMBER SPECIFYING THE        *SUTS 353
C*                            DENORMALIZED TENSION FACTOR.               *SUTS 354
C*                YP        - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF    *SUTS 355
C*                            LENGTH 2 CONTAINING SECOND DERIVATIVE      *SUTS 356
C*                            INFORMATION NECESSARY TO INTERPOLATE THE   *SUTS 357
C*                            SPLINE FUNCTION.                           *SUTS 358
C*                                                                     *SUTS 359
C*  PRECISION               - SINGLE.                                  *SUTS 360
C*                                                                     *SUTS 361
C*  REQUIRED ROUTINES       - SNHCSH,TERMS.                            *SUTS 362
```

```
C*                                                                   *SUTS 363
C*   DATE RELEASED        - MARCH 1, 1979.                           *SUTS 364
C*                                                                   *SUTS 365
C*   LANGUAGE             - FORTRAN.                                 *SUTS 366
C*                                                                   *SUTS 367
C*   SOURCE               - A. K. CLINE AND R. J. RENKA              *SUTS 368
C*                          UNIVERSITY OF TEXAS AT AUSTIN            *SUTS 369
C*                          NON-NATURAL OPTIONS ONLY                 *SUTS 370
C*                                                                   *SUTS 371
C*   LATEST REVISION      - NONE.                                    *SUTS 372
C*                                                                   *SUTS 373
C*                                                                   *SUTS 374
C*********************************************************************SUTS 375
C                                                                     SUTS 376
C              FORMAL PARAMETERS                                      SUTS 377
C                                                                     SUTS 378
      INTEGER IENDSW(2)                                               SUTS 379
C                                                                     SUTS 380
      REAL DX,END(2),SIGMAP,Y(2),YP(2)                                SUTS 381
C                                                                     SUTS 382
C              INTERNAL VARIABLES                                     SUTS 383
C                                                                     SUTS 384
      REAL DIAG,DIAGI,DIAG1,DX1,ONE,SDIAG1,SLP1,SLPN,TEMP,ZERO        SUTS 385
C                                                                     SUTS 386
C              INITIALIZE CONSTANTS                                   SUTS 387
C                                                                     SUTS 388
      DATA ZERO,ONE/0.0E0,1.0E0/                                      SUTS 389
C                                                                     SUTS 390
C              INITIALIZE  YP                                         SUTS 391
C                                                                     SUTS 392
      YP(1) = ZERO                                                    SUTS 393
      YP(2) = ZERO                                                    SUTS 394
C                                                                     SUTS 395
C              CHECK TO SEE IF EACH ENDPOINT IS EITHER NATURAL OR     SUTS 396
C              HAS A COMPUTER-CONSTRUCTED DERIVATIVE                  SUTS 397
C                                                                     SUTS 398
      IF (IENDSW(1)*IENDSW(2) .NE. 0) GO TO 9000                      SUTS 399
C                                                                     SUTS 400
C              AT LEAST ONE ENDPOINT HAS A USER-DEFINED SLOPE         SUTS 401
C              COMPUTE SOME NECESSARY CONSTANTS                       SUTS 402
C                                                                     SUTS 403
      DX1 = (Y(2) - Y(1))/DX                                          SUTS 404
      CALL TERMS(DIAG1,SDIAG1,SIGMAP,DX)                              SUTS 405
      DIAGI = ONE/DIAG1                                               SUTS 406
C                                                                     SUTS 407
C              CHECK TO SEE IF THE LEFT ENDPOINT IS NATURAL           SUTS 408
C                                                                     SUTS 409
      IF (IENDSW(1) .NE. 2) GO TO 10                                  SUTS 410
         YP(2) = (END(2) - DX1)*DIAGI                                 SUTS 411
```

```
      GO TO 9000                                                        SUTS 412
C                                                                       SUTS 413
C           CHECK TO SEE IF THE RIGHT ENDPOINT IS NATURAL               SUTS 414
C                                                                       SUTS 415
   10 IF (IENDSW(2) .NE. 2) GO TO 20                                    SUTS 416
         YP(1) = (DX1 - END(1))*DIAGI                                   SUTS 417
         GO TO 9000                                                     SUTS 418
C                                                                       SUTS 419
C           NOW AT LEAST ONE OF THE ENDPOINTS HAS A USER-SPECIFIED      SUTS 420
C           DERIVATIVE AND NEITHER IS NATURAL  -  THIS IS THE CODE      SUTS 421
C           FROM THE ORIGINAL  CURV1  WITH  N  = 2                      SUTS 422
C                                                                       SUTS 423
   20 SLP1 = END(1)                                                     SUTS 424
      IF (IENDSW(1) .EQ. 1) SLP1 = DX1                                  SUTS 425
      SLPN = END(2)                                                     SUTS 426
      IF (IENDSW(2) .EQ. 1) SLPN = DX1                                  SUTS 427
      TEMP = SDIAG1*DIAGI                                               SUTS 428
      DIAG = DIAG1 - SDIAG1*TEMP                                        SUTS 429
      YP(2) = (SLPN - DX1 - (DX1 -SLP1)*TEMP) / DIAG                    SUTS 430
      YP(1) = (DX1 - SLP1)*DIAGI  -   TEMP*YP(2)                        SUTS 431
C                                                                       SUTS 432
 9000 RETURN                                                            SUTS 433
      END                                                               SUTS 434
      FUNCTION CURVI(XL,XU,N,X,Y,YP,SIGMAP)                             SUTS 435
C                                                                       SUTS 436
C***********************************************************************SUTS 437
C*                                                                     *SUTS 438
C*                                                                     *SUTS 439
C*  PURPOSE              - THE VALUE OF  CURVI  IS THE DEFINITE INTEGRAL*SUTS 440
C*                         OF A SPLINE FUNCTION UNDER TENSION FROM  XL *SUTS 441
C*                         TO  XU  GIVEN NECESSARY INFORMATION FROM    *SUTS 442
C*                         CURV1N  OR  CURV12 .  SEE  SUTS (D1.X).     *SUTS 443
C*                                                                     *SUTS 444
C*  USE                  - CALL CURVI(XL,XU,N,X,Y,YP,SIGMAP)           *SUTS 445
C*                                                                     *SUTS 446
C*  PARAMETERS   XL      - AN INPUT REAL NUMBER SPECIFYING THE LOWER   *SUTS 447
C*                         LIMIT OF INTEGRATION.                       *SUTS 448
C*               XU      - AN INPUT REAL NUMBER SPECIFYING THE UPPER   *SUTS 449
C*                         LIMIT OF INTEGRATION.                       *SUTS 450
C*               N       - AN INPUT INTEGER SPECIFYING THE NUMBER OF   *SUTS 451
C*                         NODES.  N  MUST BE AT LEAST 2.              *SUTS 452
C*               X       - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 453
C*                         N  SPECIFYING THE X-COORDINATES OF THE NODES.*SUTS 454
C*                         X  MUST BE A STRICTLY INCREASING ARRAY.     *SUTS 455
C*               Y       - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 456
C*                         N  SPECIFYING THE Y-COORDINATES OF THE NODES.*SUTS 457
C*               YP      - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 458
C*                         N  WHICH IS OUTPUT FROM  CURV1N  OR  CURV12 .*SUTS 459
C*               SIGMAP  - AN INPUT REAL NUMBER SPECIFYING THE         *SUTS 460
```

```
C*                         DENORMALIZED TENSION FACTOR.                 *SUTS 461
C*                                                                      *SUTS 462
C*  PRECISION             - SINGLE.                                     *SUTS 463
C*                                                                      *SUTS 464
C*  REQUIRED ROUTINES     - INTRVL,SNHCSH.                              *SUTS 465
C*                                                                      *SUTS 466
C*  DATE RELEASED         - MARCH 1, 1979.                              *SUTS 467
C*                                                                      *SUTS 468
C*  LANGUAGE              - FORTRAN.                                    *SUTS 469
C*                                                                      *SUTS 470
C*  SOURCE                - A. K. CLINE AND R. J. RENKA                 *SUTS 471
C*                          UNIVERSITY OF TEXAS AT AUSTIN               *SUTS 472
C*                                                                      *SUTS 473
C*  LATEST REVISION       - NONE.                                       *SUTS 474
C*                                                                      *SUTS 475
C*                                                                      *SUTS 476
C***********************************************************************SUTS 477
C                                                                        SUTS 478
C               FORMAL PARAMETERS                                        SUTS 479
C                                                                        SUTS 480
      INTEGER N                                                          SUTS 481
C                                                                        SUTS 482
      REAL SIGMAP,X(N),XL,XU,Y(N),YP(N)                                  SUTS 483
C                                                                        SUTS 484
C               INTERNAL VARIABLES                                       SUTS 485
C                                                                        SUTS 486
      INTEGER I,IL,ILM1,ILP1,IM1,IU,IUM1                                 SUTS 487
C                                                                        SUTS 488
      REAL CL1,CL2,CS,CU1,CU2,C1,C2,DELI,DELL1,DELL2,DELS,DELSI,DELU1    SUTS 489
      REAL DELU2,DEL1,DEL2,DUMMY,HALF,ONE,SIGCUBE,SIGDELS,SIXTH,SS       SUTS 490
      REAL SSIGN,SUM,THIRD,TWELFTH,T1,T2,XXL,XXU,ZERO                    SUTS 491
C                                                                        SUTS 492
C               INITIALIZE CONSTANTS                                     SUTS 493
C                                                                        SUTS 494
      DATA ZERO,ONE,HALF/0.0E0,1.0E0,5.0E-1/                             SUTS 495
      DATA THIRD,SIXTH,TWELFTH/                                          SUTS 496
     $17165252525252525253B,17155252525252525253B,17145252525252525253B/SUTS 497
C           THIRD = 1./3.                                                SUTS 498
C           SIXTH = 1./6.                                                SUTS 499
C           TWELFTH = 1./12.                                             SUTS 500
C                                                                        SUTS 501
C               CONSTRUCT THE STATEMENT FUNCTION  TERM                   SUTS 502
C               TERM  COMPUTES COEFFICIENTS FOR ANTI-DERIVATIVES         SUTS 503
C                                                                        SUTS 504
      TERM(CMM1,CMM2,T) = (CMM1 - CMM2 - SIGMAP*T*SS) /                  SUTS 505
     $                    (SIGCUBE*(SS + SIGDELS))                       SUTS 506
      SIGCUBE = SIGMAP*SIGMAP*SIGMAP                                     SUTS 507
C                                                                        SUTS 508
C               DETERMINE ACTUAL UPPER AND LOWER LIMITS                  SUTS 509
```

```
C
      XXL = XL
      XXU = XU
      SSIGN = ONE
      IF (XXL .LT. XXU) GO TO 10
      XXL = XXU
      XXU = XL
      SSIGN = -ONE
      IF (XXU .GT. XXL) GO TO 10
C
C             RETURN ZERO IF  XL  =  XU
C
         CURVI = ZERO
         GO TO 9000
C
C             SEARCH FOR THE PROPER SUBINTERVALS
C             X(ILM1)  .LE.  XXL  .LE.  X(IL)
C
   10 ILM1 = INTRVL (XXL,X,N)
      IL = ILM1 + 1
C
C             X(IUM1)  .LE.  XXU  .LE.  X(IU)
C
      IUM1 = INTRVL (XXU,X,N)
      IU = IUM1 + 1
C
C             BRANCH IF  XXU  AND  XXL  ARE IN THE SAME SUBINTERVAL
C
      IF (IL .EQ. IU) GO TO 70
      SUM = ZERO
C
C             BRANCH IF  XXL  =  X(IL)
C
      IF (XXL .EQ. X(IL)) GO TO 20
C
C             INTEGRATE FROM  XXL  TO  X(IL)
C             FOR THIS CODE  XXL  .LT.  XX(IL)  .LE.  XX(IUM1)
C
      DEL1 = XXL - X(ILM1)
      DEL2 = X(IL) - XXL
      DELS = DEL1 + DEL2
      DELSI = HALF/DELS
      T1 = (DEL1 + DELS)*DEL2*DELSI
      T2 = DEL2*DEL2*DELSI
      SUM = T1*Y(IL) + T2*Y(ILM1)
      IF (SIGMAP .EQ. ZERO) SUM = SUM - T1*T1*DELS*YP(IL)*SIXTH
     $   - T2 * (DEL1*(DEL2 + DELS) + DELS*DELS) * YP(ILM1)*TWELFTH
      IF (SIGMAP .EQ. ZERO) GO TO 20
C
```

```
C            THIS CODE IS FOR A SPLINE UNDER TENSION
C
      SIGDELS = SIGMAP*DELS
      CALL SNHCSH(DUMMY,C1,SIGMAP*DEL1,2)
      CALL SNHCSH(DUMMY,C2,SIGMAP*DEL2,2)
      CALL SNHCSH(SS,CS,SIGDELS,3)
      SUM = SUM + TERM(CS,C1,T1)*YP(IL) + TERM(C2,ZERO,T2)*YP(ILM1)
C
C            BRANCH IF  X(IL)  =  X(IUM1)
C
   20 IF (IL .EQ. IUM1) GO TO 60
C
C            INTEGRATE OVER THE INTERIOR INTERVALS WITH A LOOP
C            FOR THIS CODE  X(IL)  <  X(IUM1)
C
    ILP1 = IL + 1
    IF (SIGMAP .NE. ZERO) GO TO 40
C
C            THIS CODE IS FOR A CUBIC SPLINE
C
      DO 30 I=ILP1,IUM1
         IM1 = I - 1
         DELS = (X(I) - X(IM1))*HALF
         SUM = SUM + (Y(I) + Y(IM1))*DELS - (YP(I) +YP(IM1))*
     $                DELS*DELS*DELS*THIRD
   30 CONTINUE
      GO TO 60
C
C            THIS CODE IS FOR A SPLINE UNDER TENSION
C
   40 DO 50 I=ILP1,IUM1
      IM1 = I - 1
      DELS = X(I) - X(IM1)
      SIGDELS = SIGMAP*DELS
      CALL SNHCSH(SS,CS,SIGDELS,3)
      SUM = SUM + (Y(I) + Y(IM1))*DELS*HALF + (YP(I) + YP(IM1))*
     $           (CS - SS*SIGDELS*HALF)/(SIGCUBE*(SS + SIGDELS))
   50 CONTINUE
C
C            BRANCH IF  X(IU-1)  =  XXU
C
   60 IF (XXU .EQ. X(IUM1)) GO TO 80
C
C            INTEGRATE FROM  X(IU-1)  TO  XXU
C            FOR THIS CODE  X(IU-1)  <  XXU
C
    DEL1 = XXU - X(IUM1)
    DEL2 = X(IU) - XXU
    DELS = X(IU) - X(IUM1)
```

```
      DELSI = HALF/DELS
      T1 = DEL1*DEL1*DELSI
      T2 = (DEL2 + DELS)*DEL1*DELSI
      SUM = SUM + T1*Y(IU) + T2*Y(IUM1)
      IF (SIGMAP .EQ. ZERO) SUM = SUM - T1*(DEL2*(DEL1 + DELS) +
     $     DELS*DELS)* YP(IU)*TWELFTH - T2*T2*DELS*YP(IUM1)*SIXTH
      IF (SIGMAP .EQ. ZERO) GO TO 80
         SIGDELS = SIGMAP*DELS
         CALL SNHCSH (DUMMY,C1,SIGMAP*DEL1,2)
         CALL SNHCSH (DUMMY,C2,SIGMAP*DEL2,2)
         CALL SNHCSH (SS,CS,SIGDELS,3)
         SUM = SUM + TERM(C1,ZERO,T1)*YP(IU) + TERM(CS,C2,T2)*YP(IUM1)
         GO TO 80
C
C              INTEGRATE FROM  XXL  TO  XXU
C              FOR THIS CODE  X(IUM1)  .LE.  XXL  .LT.  XXU  .LE.  X(IU)
C
   70 DELU1 = XXU - X(IUM1)
      DELU2 = X(IU) - XXU
      DELL1 = XXL - X(IUM1)
      DELL2 = X(IU) - XXL
      DELS = X(IU) - X(IUM1)
      DELI = XXU - XXL
      DELSI = HALF/DELS
      T1 = (DELU1 + DELL1)*DELI*DELSI
      T2 = (DELU2 + DELL2)*DELI*DELSI
      SUM = T1*Y(IU) + T2*Y(IUM1)
      IF (SIGMAP .EQ. ZERO) SUM = SUM - T1*(DELU2*(DELS + DELU1) + DELL2
     $      *(DELS + DELL1))*YP(IU)*TWELFTH - T2*(DELL1*(DELS + DELL2)
     $      + DELU1*(DELS + DELU2))*YP(IUM1)*TWELFTH
      IF (SIGMAP .EQ. ZERO) GO TO 80
C
C             THIS CODE IS FOR A SPLINE UNDER TENSION
C
         SIGDELS = SIGMAP*DELS
         CALL SNHCSH (DUMMY,CU1,SIGMAP*DELU1,2)
         CALL SNHCSH (DUMMY,CU2,SIGMAP*DELU2,2)
         CALL SNHCSH (DUMMY,CL1,SIGMAP*DELL1,2)
         CALL SNHCSH (DUMMY,CL2,SIGMAP*DELL2,2)
         CALL SNHCSH (SS,DUMMY,SIGDELS,-1)
         SUM = SUM + TERM(CU1,CL1,T1)*YP(IU) + TERM(CL2,CU2,T2)*YP(IUM1)
C
C             CORRECT SIGN AND RETURN
C
   80 CURVI = SSIGN*SUM
C
 9000 RETURN
      END
      SUBROUTINE CEEZ (DEL1,DEL2,SIGMA,C1,C2,C3,N)
```

```
C                                                                       SUTS 657
C***********************************************************************SUTS 658
C*                                                                     *SUTS 659
C*                                                                     *SUTS 660
C*   PURPOSE               - DETERMINE COEFFICIENTS  C1 ,  C2 , AND  C3   *SUTS 661
C*                           USED TO DETERMINE ENDPOINT SLOPES.  GIVEN THE*SUTS 662
C*                           POINTS  (X1,Y1) ,  (X2,Y2) , AND  (X3,Y3) ,  *SUTS 663
C*                           THE ENDPOINT SLOPE IS  C1*Y1 + C2*Y2 + C3*Y3 *SUTS 664
C*                           IF  N  > 2 AND  C1*Y1 + C2*Y2  IF  N  = 2.   *SUTS 665
C*                                                                     *SUTS 666
C*   USE                   - CALL CEEZ(DEL1,DEL2,SIGMA,C1,C2,C3,N)      *SUTS 667
C*                                                                     *SUTS 668
C*   PARAMETERS    DEL1    - AN INPUT REAL NUMBER =  X2 - X1.  DEL1  > 0. *SUTS 669
C*                 DEL2    - AN INPUT REAL NUMBER =  X3 - X1.  IF  N  = 2,*SUTS 670
C*                           THEN  DEL2  MAY BE A DUMMY PARAMETER.  DEL2  *SUTS 671
C*                           > 0.                                       *SUTS 672
C*                 SIGMA   - AN INPUT REAL NUMBER SPECIFYING THE TENSION  *SUTS 673
C*                           FACTOR.                                    *SUTS 674
C*                 C1      - AN OUTPUT REAL NUMBER.                     *SUTS 675
C*                 C2      - AN OUTPUT REAL NUMBER.                     *SUTS 676
C*                 C3      - AN OUTPUT REAL NUMBER.  IF  N  = 2, THEN  C3 *SUTS 677
C*                           IS A DUMMY PARAMETER.                      *SUTS 678
C*                 N       - AN INPUT INTEGER SPECIFYING THE NUMBER OF  *SUTS 679
C*                           NODES.                                     *SUTS 680
C*                                                                     *SUTS 681
C*   PRECISION             - SINGLE.                                    *SUTS 682
C*                                                                     *SUTS 683
C*   REQUIRED ROUTINES     - NONE.                                      *SUTS 684
C*                                                                     *SUTS 685
C*   DATE RELEASED         - MARCH 1, 1979.                             *SUTS 686
C*                                                                     *SUTS 687
C*   LANGUAGE              - FORTRAN.                                   *SUTS 688
C*                                                                     *SUTS 689
C*   SOURCE                - A. K. CLINE AND R. J. RENKA                *SUTS 690
C*                           UNIVERSITY OF TEXAS AT AUSTIN              *SUTS 691
C*                                                                     *SUTS 692
C*   LATEST REVISION       - NONE.                                      *SUTS 693
C*                                                                     *SUTS 694
C*                                                                     *SUTS 695
C***********************************************************************SUTS 696
C                                                                       SUTS 697
C             FORMAL PARAMETERS                                         SUTS 698
C                                                                       SUTS 699
      INTEGER N                                                         SUTS 700
C                                                                       SUTS 701
      REAL C1,C2,C3,DEL1,DEL2,SIGMA                                     SUTS 702
C                                                                       SUTS 703
C             INTERNAL VARIABLES                                        SUTS 704
C                                                                       SUTS 705
```

```
      REAL COSHM1,DEL,DENOM,ONE,SIN,TWO,ZERO                            SUTS 706
C                                                                       SUTS 707
C              INITIALIZE CONSTANTS                                     SUTS 708
C                                                                       SUTS 709
      DATA ZERO,ONE,TWO/0.0E0,1.0E0,2.0E0/                              SUTS 710
C                                                                       SUTS 711
C              TEST WHETHER  N  = 2                                     SUTS 712
      IF (N .EQ. 2) GO TO 20                                            SUTS 713
C                                                                       SUTS 714
C              INITIALIZE THE VARIABLE  DEL                             SUTS 715
C                                                                       SUTS 716
         DEL = DEL2 - DEL1                                              SUTS 717
C                                                                       SUTS 718
C              TEST WHETHER  SIGMA  = ZERO                              SUTS 719
C                                                                       SUTS 720
         IF (SIGMA .NE. ZERO) GO TO 10                                  SUTS 721
C                                                                       SUTS 722
C              THIS CODE IS FOR A STANDARD CUBIC SPLINE                 SUTS 723
C                                                                       SUTS 724
            C1 = -(DEL1+DEL2)/(DEL1*DEL2)                               SUTS 725
            C2 = DEL2/(DEL1*DEL)                                        SUTS 726
            C3 = -DEL1/(DEL2*DEL)                                       SUTS 727
            GO TO 9000                                                  SUTS 728
C                                                                       SUTS 729
C              THIS CODE IS FOR A SPLINE UNDER TENSION                  SUTS 730
C                                                                       SUTS 731
   10    CALL SNHCSH(DUMMY,COSHM1,SIGMA*DEL1,1)                         SUTS 732
         CALL SNHCSH(DUMMY,COSHM2,SIGMA*DEL2,1)                         SUTS 733
         SIN = TWO * SINH(SIGMA*(DEL2+DEL1)/TWO) * SINH(SIGMA*DEL/TWO)  SUTS 734
         DENOM = ONE / (COSHM1*DEL - DEL1*SIN)                          SUTS 735
         C1 = SIN * DENOM                                               SUTS 736
         C2 = -COSHM2 * DENOM                                           SUTS 737
         C3 = COSHM1 * DENOM                                            SUTS 738
         GO TO 9000                                                     SUTS 739
C                                                                       SUTS 740
C              TWO COEFFICIENTS                                         SUTS 741
C                                                                       SUTS 742
   20 C1 = -ONE/DEL1                                                    SUTS 743
      C2 = -C1                                                          SUTS 744
 9000 RETURN                                                            SUTS 745
C                                                                       SUTS 746
      END                                                               SUTS 747
      SUBROUTINE TERMS(DIAG,SDIAG,SIGMA,DEL)                            SUTS 748
C                                                                       SUTS 749
C***********************************************************************SUTS 750
C*                                                                     *SUTS 751
C*                                                                     *SUTS 752
C*  PURPOSE                - COMPUTE THE DIAGONAL AND SUPERDIAGONAL TERMS *SUTS 753
C*                           OF THE TRIDIAGONAL LINEAR SYSTEM ASSOCIATED  *SUTS 754
```

```
C*                        WITH SPLINE UNDER TENSION INTERPOLATION.     *SUTS 755
C*                                                                     *SUTS 756
C*  USE                 - CALL TERMS(DIAG,SDIAG,SIGMA,DEL)             *SUTS 757
C*                                                                     *SUTS 758
C*  PARAMETERS   DIAG   - AN OUTPUT REAL NUMBER CONTAINING THE DIAGONAL*SUTS 759
C*                        TERM.                                        *SUTS 760
C*              SDIAG   - AN OUTPUT REAL NUMBER CONTAINING THE         *SUTS 761
C*                        SUPERDIAGONAL TERM.                          *SUTS 762
C*              SIGMA   - AN INPUT REAL NUMBER SPECIFYING THE TENSION  *SUTS 763
C*                        FACTOR.                                      *SUTS 764
C*                DEL   - AN INPUT REAL NUMBER SPECIFYING THE STEP     *SUTS 765
C*                        SIZE.                                        *SUTS 766
C*                                                                     *SUTS 767
C*  PRECISION           - SINGLE.                                      *SUTS 768
C*                                                                     *SUTS 769
C*  REQUIRED ROUTINES   - NONE.                                        *SUTS 770
C*                                                                     *SUTS 771
C*  DATE RELEASED       - MARCH 1, 1979.                               *SUTS 772
C*                                                                     *SUTS 773
C*  SOURCE              - A. K. CLINE AND R. J. RENKA                  *SUTS 774
C*                        UNIVERSITY OF TEXAS AT AUSTIN                *SUTS 775
C*                                                                     *SUTS 776
C*  LATEST REVISION     - NONE.                                        *SUTS 777
C*                                                                     *SUTS 778
C*                                                                     *SUTS 779
C***********************************************************************SUTS 780
C                                                                       SUTS 781
C                                                                       SUTS 782
C               FORMAL PARAMETERS                                       SUTS 783
C                                                                       SUTS 784
      REAL DEL,DIAG,SDIAG,SIGMA                                         SUTS 785
C                                                                       SUTS 786
C               INTERNAL VARIABLES                                      SUTS 787
C                                                                       SUTS 788
      REAL COSHM,DENOM,SIGDEL,SINHM,SIXTH,THIRD                         SUTS 789
C                                                                       SUTS 790
C               INITIALIZE CONSTANTS                                    SUTS 791
C                                                                       SUTS 792
      DATA ZERO,SIXTH,THIRD/                                            SUTS 793
     $0.0E0,17155252525252525253B,17165252525252525253B/                SUTS 794
C         SIXTH = 1./6.                                                 SUTS 795
C         TWELFTH = 1./12.                                              SUTS 796
C                                                                       SUTS 797
C               TEST WHETHER  SIGMA  IS ZERO                            SUTS 798
C                                                                       SUTS 799
      IF (SIGMA .NE. ZERO) GO TO 10                                     SUTS 800
C                                                                       SUTS 801
C               THIS CODE IS FOR THE STANDARD CUBIC SPLINE              SUTS 802
C                                                                       SUTS 803
```

```
      DIAG = DEL*THIRD
      SDIAG = DEL*SIXTH
      GO TO 9000
C
C             THIS CODE IS FOR A SPLINE UNDER NON-ZERO TENSION
C
   10 SIGDEL = SIGMA*DEL
      CALL SNHCSH(SINHM,COSHM,SIGDEL,0)
      DENOM = DEL / ((SINHM + SIGDEL) * SIGDEL * SIGDEL)
      DIAG = DENOM * (SIGDEL*COSHM - SINHM)
      SDIAG = DENOM * SINHM
C
 9000 RETURN
      END
      SUBROUTINE SNHCSH(SINHM,COSHM,X,ISW)
C
C***********************************************************************
C*                                                                     *
C*                                                                     *
C*  PURPOSE              - APPROXIMATE  SINHM(X) = SINH(X) - X         *
C*                                      COSHM(X) = COSH(X) - 1         *
C*                         AND         COSHMM(X) = COSH(X) - 1 - X*X/2 *
C*                         WITH RELATIVE ERROR < 3.42E-14              *
C*                                                                     *
C*  USE                  - CALL SNHCSH(SINHM,COSHM,X,ISW)              *
C*                                                                     *
C*  PARAMETERS   SINHM   - AN OUTPUT REAL NUMBER CONTAINING  SINHM(X)  *
C*                         IF  ISW  = -1, 0, OR 3.  OTHERWISE  SINHM  IS*
C*                         UNCHANGED UPON RETURN.                      *
C*               COSHM   - AN OUTPUT REAL NUMBER CONTAINING  COSHM(X)  *
C*                         IF  ISW  = 0 OR 1 AND CONTAINING  COSHMM(X) *
C*                         ISW  = 2 OR 3.  OTHERWISE  COSHM IS UNCHANGED*
C*                         UPON RETURN.                                *
C*               X       - AN INPUT REAL NUMBER CONTAINING THE         *
C*                         INDEPENDENT VARIABLE.                       *
C*               ISW     - AN INPUT INTEGER SPECIFYING THE FUNCTION    *
C*                         DESIRED.                                    *
C*                         = -1  (ONLY)  SINHM  IS DESIRED.            *
C*                         =  0  SINHM  AND  COSHM  ARE DESIRED.       *
C*                         =  1  (ONLY)  COSHM  IS DESIRED.            *
C*                         =  2  (ONLY)  COSHMM  IS DESIRED.           *
C*                         =  3  SINHM  AND  COSHMM  ARE DESIRED.      *
C*                                                                     *
C*  PRECISION            - SINGLE.                                     *
C*                                                                     *
C*  REQUIRED ROUTINES    - NONE.                                       *
C*                                                                     *
C*  DATE RELEASED        - MARCH 1, 1979.                              *
C*                                                                     *
```

```
C*   LANGUAGE             - FORTRAN.                                    *SUTS 853
C*                                                                      *SUTS 854
C*   SOURCE               - A. K. CLINE AND R. J. RENKA                 *SUTS 855
C*                          UNIVERSITY OF TEXAS AT AUSTIN               *SUTS 856
C*                                                                      *SUTS 857
C*   LATEST REVISION      - NONE.                                       *SUTS 858
C*                                                                      *SUTS 859
C*                                                                      *SUTS 860
C***********************************************************************SUTS 861
C                                                                        SUTS 862
C                FORMAL PARAMETERS                                       SUTS 863
C                                                                        SUTS 864
      INTEGER ISW                                                        SUTS 865
C                                                                        SUTS 866
      REAL COSHM,SINHM,X                                                 SUTS 867
C                                                                        SUTS 868
C                INTERNAL PARAMETERS                                     SUTS 869
C                                                                        SUTS 870
      REAL AX,CP1,CP2,CP3,CP4,CQ1,EXPX,SP1,SP2,SP3,SP4,SQ1,XS,XX,ZP1,ZP2SUTS 871
      REAL ZP3,ZQ1,ZQ2,ZQ3,ZQ4                                           SUTS 872
C                                                                        SUTS 873
C                                                                        SUTS 874
      DATA SP4/4.50217693381333E-08/,                                    SUTS 875
     *     SP3/8.95278544216390E-06/,                                    SUTS 876
     *     SP2/8.72048976791502E-04/,                                    SUTS 877
     *     SP1/4.36314556981690E-02/,                                    SUTS 878
     *     SQ1/-6.36854430175110E-03/                                    SUTS 879
      DATA CP4/1.78419567490190E-07/,                                    SUTS 880
     *     CP3/2.87277229799044E-05/,                                    SUTS 881
     *     CP2/2.15151519902028E-03/,                                    SUTS 882
     *     CP1/7.58181822756256E-02/,                                    SUTS 883
     *     CQ1/-7.51515105679867E-03/                                    SUTS 884
      DATA ZP3/5.59297116264720E-07/,                                    SUTS 885
     *     ZP2/1.77943490308894E-04/,                                    SUTS 886
     *     ZP1/1.69800461694792E-02/,                                    SUTS 887
     *     ZQ4/1.33412535492375E-09/,                                    SUTS 888
     *     ZQ3/-5.80858944138663E-07/,                                   SUTS 889
     *     ZQ2/1.27814964403863E-04/,                                    SUTS 890
     *     ZQ1/-1.63532871439181E-02/                                    SUTS 891
      XX = X                                                             SUTS 892
      AX = ABS(XX)                                                       SUTS 893
      XS = XX*XX                                                         SUTS 894
      IF ((AX .GE. 2.70) .OR. (AX .GE. 1.15 .AND.                        SUTS 895
     *     ISW .NE. 2)) EXPX = EXP(AX)                                   SUTS 896
C                                                                        SUTS 897
C                APPROXIMATE  SINHM                                      SUTS 898
C                                                                        SUTS 899
      IF (ISW .EQ. 1 .OR. ISW .EQ. 2) GO TO 2                            SUTS 900
      IF (AX .GE. 1.15) GO TO 1                                          SUTS 901
```

```
      SINHM = (((((SP4*XS+SP3)*XS+SP2)*XS+SP1)*XS+1.)*XS*XX)              SUTS 902
     *        /((SQ1*XS+1.)*6.)                                           SUTS 903
      GO TO 2                                                             SUTS 904
    1 SINHM = -(((1./EXPX+AX)+AX)-EXPX)/2.                                SUTS 905
      IF (XX .LT. 0.) SINHM = -SINHM                                      SUTS 906
C                                                                         SUTS 907
C               APPROXIMATE  COSHM                                        SUTS 908
C                                                                         SUTS 909
    2 IF (ISW .NE. 0 .AND. ISW .NE. 1) GO TO 4                            SUTS 910
      IF (AX .GE. 1.15) GO TO 3                                           SUTS 911
      COSHM = (((((CP4*XS+CP3)*XS+CP2)*XS+CP1)*XS+1.)*XS)                 SUTS 912
     *        /((CQ1*XS+1.)*2.)                                           SUTS 913
      GO TO 4                                                             SUTS 914
    3 COSHM = ((1./EXPX-2.)+EXPX)/2.                                      SUTS 915
C                                                                         SUTS 916
C               APPROXIMATE  COSHMM                                       SUTS 917
C                                                                         SUTS 918
    4 IF (ISW .LE. 1) RETURN                                              SUTS 919
      IF (AX .GE. 2.70) GO TO 5                                           SUTS 920
      COSHM = ((((ZP3*XS+ZP2)*XS+ZP1)*XS+1.)*XS*XS)/(((((ZQ4              SUTS 921
     *        *XS+ZQ3)*XS+ZQ2)*XS+ZQ1)*XS+1.)*24.)                        SUTS 922
      RETURN                                                              SUTS 923
    5 COSHM = (((1./EXPX-2.)-XS)+EXPX)/2.                                 SUTS 924
      RETURN                                                              SUTS 925
      FND                                                                 SUTS 926
      FUNCTION INTRVL(T,X,N)                                              SUTS 927
C                                                                         SUTS 928
C***********************************************************************SUTS 929
C*                                                                     *SUTS 930
C*                                                                     *SUTS 931
C*   PURPOSE             - DETERMINE THE INDEX OF THE INTERVAL         *SUTS 932
C*                         (DETERMINED BY A GIVEN INCREASING SEQUENCE) *SUTS 933
C*                         IN WHICH A GIVEN VALUE LIES.                *SUTS 934
C*                                                                     *SUTS 935
C*   USE                 - I = INTRVL(T,X,N)                           *SUTS 936
C*                                                                     *SUTS 937
C*   PARAMETERS    T     - AN INPUT REAL NUMBER SPECIFYING THE         *SUTS 938
C*                         GIVEN VALUE.                                *SUTS 939
C*                 X     - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH*SUTS 940
C*                         N  SPECIFYING THE INCREASING SEQUENCE.  X   *SUTS 941
C*                         MUST BE STRICTLY INCREASING.                *SUTS 942
C*                 N     - AN INPUT INTEGER SPECIFYING THE LENGTH OF   *SUTS 943
C*                         X .  N  > 1.                                *SUTS 944
C*                                                                     *SUTS 945
C*   OUTPUT        I     - IF  T  .LE.  X(2) ,  I  = 1.                *SUTS 946
C*                         IF  T  .GE.  X(N-1) ,  I  =  N - 1 .        *SUTS 947
C*                         OTHERWISE,  X(I)  .LE.  T  .LE.  X(I+1)     *SUTS 948
C*                                                                     *SUTS 949
C*   PRECISION           - SINGLE.                                     *SUTS 950
```

```
C*                                                                     *SUTS 951
C*  REQUIRED ROUTINES    - NONE.                                       *SUTS 952
C*                                                                     *SUTS 953
C*  DATE RELEASED        - MARCH 1, 1979.                              *SUTS 954
C*                                                                     *SUTS 955
C*  SOURCE               - A. K. CLINE AND R. J. RENKA                 *SUTS 956
C*                         UNIVERSITY OF TEXAS AT AUSTIN               *SUTS 957
C*                                                                     *SUTS 958
C*  LATEST REVISION      - NONE.                                       *SUTS 959
C*                                                                     *SUTS 960
C*                                                                     *SUTS 961
C***********************************************************************SUTS 962
C                                                                       SUTS 963
C                                                                       SUTS 964
C               FORMAL PARAMETERS                                       SUTS 965
C                                                                       SUTS 966
      INTEGER N                                                         SUTS 967
C                                                                       SUTS 968
      REAL T,X(N)                                                       SUTS 969
C                                                                       SUTS 970
C               INTERNAL VARIABLES                                      SUTS 971
C                                                                       SUTS 972
      INTEGER IH,IL                                                     SUTS 973
C                                                                       SUTS 974
      REAL TT                                                           SUTS 975
C                                                                       SUTS 976
      TT = T                                                            SUTS 977
      IF (TT .LE. X(2)) GO TO 4                                         SUTS 978
        IF (TT .GE. X(N-1)) GO TO 5                                     SUTS 979
          IL = 2                                                        SUTS 980
          IH = N-1                                                      SUTS 981
C                                                                       SUTS 982
C               INTERPOLATE LINEARLY FOR I                              SUTS 983
C                                                                       SUTS 984
    1     I = IL+IFIX(FLOAT(IH-IL)*(TT-X(IL))/(X(IH)-X(IL)))            SUTS 985
          IF (TT .LT. X(I)) GO TO 2                                     SUTS 986
            IF (TT .LE. X(I+1)) GO TO 3                                 SUTS 987
C                                                                       SUTS 988
C               I  IS TOO SMALL - ADJUST AND TRY AGAIN                  SUTS 989
C                                                                       SUTS 990
              IL = I+1                                                  SUTS 991
              GO TO 1                                                   SUTS 992
C                                                                       SUTS 993
C               I  IS TOO LARGE - ADJUST AND TRY AGAIN                  SUTS 994
C                                                                       SUTS 995
    2     IH = I                                                        SUTS 996
          GO TO 1                                                       SUTS 997
C                                                                       SUTS 998
    3       INTRVL = I                                                  SUTS 999
```

```
C             I  IS JUST RIGHT - RETURN                                 SUTS1000
C                                                                       SUTS1001
               RETURN                                                   SUTS1002
C                                                                       SUTS1003
C             LEFT END                                                  SUTS1004
C                                                                       SUTS1005
    4 INTRVL = 1                                                        SUTS1006
      RETURN                                                            SUTS1007
C                                                                       SUTS1008
C             RIGHT END                                                 SUTS1009
C                                                                       SUTS1010
    5   INTRVL = N-1                                                    SUTS1011
        RETURN                                                          SUTS1012
C                                                                       SUTS1013
      END                                                               SUTS1014
```

REFERENCES

1. Margason, R. J. and Lamar, J. E.: Vortex-Lattice FORTRAN Program for Estimating Subsonic Aerodynamic Characteristics of Complex Planforms. NASA TN D-6142, Feb. 1971.

2. Lamar, J. E., and Gloss, B. B. : Subsonic Aerodynamic Characteristics of Interacting Lifting Surfaces with Separated Flow around Sharp Edges Predicted by a Vortex-Lattice Method. NASA TN D-7921, Sept. 1975.

3. Lamar, J. E. and Frink, N. T. : Experimental and Analytic Study of the Longitudinal Aerodynamic Characteristics of Analytically and Empirically Designed Strake-Wing Configurations at Subcritical Speeds. NASA TP-1803, June 1981.

4. Lamar, J.E., and Herbert, H.E. : Production Version of the Extended NASA - Langley Vortex Lattice FORTRAN Computer Program, Vol 1. NASA TM 83303, April, 1982.

5. Wirth, N. : Algorithms + Data Structures = Programs. Prentice-Hall,1976.

6. Akima, H. : A Method of Bivariate Interpolation And Smooth Surface Fitting For Irregularly Distributed Data Points. ACM Trans. Math Software 4, 2(June,1978), pp. 148-159.

7. Subprogram Library. Vol. II of the Computer Programing Manual. Langley Research Center, NASA, February, 1982.

8. CYBER Loader Version 1 Reference Manual, NOS 1.4. Control Data Corporation, June, 1981.

9. Special Capabilities, Vol. IV of the Computer Programing Manual, Langley Research Center, NASA, February, 1982.

```
*
*            SEGMENTATION DIRECTIVES FOR THE
*            VORTEX LATTICE FORTRAN PROGRAM
*
*            H. E. HERBERT
*            COMPUTER SCIENCES CORPORATION
*            HAMPTON, VA.
*            JULY, 1981
*
*
 ROOT TREE WINGAL-(GEOMTRY,MATXSOL,AERODYN,CDRAGNF,TIPSUCT,VORTEX,CNLONG
,)
         INCLUDE  WINGAL,INFSUB,LOADING,FTLUP,READIN
         GLOBAL   ALL,TOTHREE,THREFOR,ONETHRE,MAINONE,CCRRDD,INSUB23
*
*
 GEOMTRY INCLUDE  GEOMTRY,PLANPLT
*
 MATXSOL INCLUDE  MATXSOL,GIVENS,BLOCKR,TRIANG,SOLVER,BUFFIN
*
 AERODYN INCLUDE  AERODYN,FLOWFL,CDICLS,HEAPSRT,SIFT
*
 CDRAGNF INCLUDE  CDRAGNF
*
 TIPSUCT INCLUDE  TIPSUCT,WRTANS
*
 VORTEX  INCLUDE  VORTEX
*
 CNLONG  INCLUDE  CNLONG,INTERP,IQHSCV,IQHSD,IQHSE,IQHSF,IQHSG
 CNLONG  INCLUDE  IQHSH,UERTST,UGETIO,SUTS,CURV1N,CURV12,CURVI
 CNLONG  INCLUDE  CEEZ,TERMS,SNHCSH,INTRVL
*
*
         END
```

FIGURE 1 - SEGMENTATION DIRECTIVES

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM 83304		

4. Title and Subtitle	5. Report Date
Production Version of the Extended NASA-Langley Vortex Lattice FORTRAN Computer Program. Vol. II Source Code	April 1982
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Henry E. Herbert John E. Lamar	505-31-43-03
	10. Work Unit No.
9. Performing Organization Name and Address Langley Research Center Hampton, VA 23665	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Henry E. Herbert, Computer Science Corporation, Hampton, VA 23665
John E. Lamar, Langley Research Center

16. Abstract

This document presents the source code for the latest production version, MARK IV, of the NASA-Langley Vortex Lattice Computer Program. All viable subcritical aerodynamic features of previous versions have been retained. This version extends the previously documented program capabilities to four planforms, 400 panels, and enables the user to obtain vortex-flow aerodynamics on cambered planforms, flow-field properties off the configuration in attached flow, and planform longitudinal load distributions.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Vortex Lattice Method Segmentation Directives	Unclassified [redacted] Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	150	A07